Repairing Redistricting:

Using an Integer Linear Programming Model to

Optimize Fairness in Congressional Districts

A thesis presented to

the Honors Tutorial College

Ohio University

In partial fulfillment

of the requirements for graduation

from the Honors Tutorial College

with the degree of

Bachelor of Science

Benjamin A. Carman

May 2021

This thesis titled

Repairing Redistricting:

Using an Integer Linear Programming Model to

Optimize Fairness in Congressional Districts

by

BENJAMIN A. CARMAN

has been approved by

the Honors Tutorial College

and the Department of Mathematics by

_____

Dr. Vardges Melkonian

Associate Professor, Mathematics

Thesis Advisor

_____

Dr. Alexei Davydov

Director of Studies, Mathematics, Honors Tutorial College

_____

Dr. Donal Skinner

Dean, Honors Tutorial College

Abstract

Historically, redistricting has been a process ridden with political manipulation in which politicians “gerrymander” districts to achieve a competitive advantage in future elections. However, the process of redistricting can be aided significantly by mathematical models that prioritize key characteristics of a “fair” district. This paper details one such integer linear programming model implemented in AMPL that ensures just that—a fair district. To ensure fairness, the model produces districts that reflect the political distribution of the state, with no party favored to win more districts than their share of the statewide vote. At the same time, the model prioritizes even population distribution while constraining for contiguous and compact districts. This model is tested and evaluated on data from the state of Ohio and details some possible variations and future directions that allow the model to adapt to other states and goals.

Acknowledgments

I would like to offer my deepest thanks to my HTC thesis advisor for this project, Dr. Vardges Melkonian. I first learned the skills that laid the groundwork for this thesis in a tutorial with him during the Fall of 2018 on Operations Research and started working on this project that same semester. I've always been passionate about social and political issues and thanks to his encouragement of hand-on self-discovery through an open-ended final project, I was able to connect that passion to math and computing. This allowed me to not only deeply explore mathematical modeling and optimization, but to apply that to a fascinating project and truly it my own. That experience has been opening up doors for me to continuing applying math and computing on interesting, impactful problems ever since. Dr. Melkonian has been nothing short of a dedicated teacher, gracious advisor, and inspiring mentor since that fall of my sophomore year and for that I am truly grateful.

I would also like to thank the Honors Tutorial College for making this thesis possible by allowing me to have space in my curriculum to discover and explore my interests and passions in math and computer science. I have had a challenging and engaging curriculum filled with opportunities for experiential learning thanks to HTC.

Finally, I would like to thank the Metric Gerrymandering and Geometry Group (MGGG) at Tufts University. After starting this project, I became deeply interested in the intersection of STEM, policy, and redistricting. I've since had the opportunity to train with MGGG over the summer on exactly this as well as on many of the data wrangling skills I needed to make this project possible. I have been collaborating with MGGG on a variety of projects ever since and been grateful to continue learning along the way.

Table of Contents

Page

List of Figures

List of Tables

List of Maps

Page

Chapter 1: Introduction

Redistricting is the process by which new lines are drawn to create districts for the U.S. House of Representatives and state legislatures. This legally required process happens in each state every ten years following the recently completed census [1]. As new data suggests significant population changes, new districts are required to ensure equal representation among states and among districts within the states. Historically, however, redistricting has rarely been a process by which "equal representation" is a priority. In fact, many states have politically biased legislators and state officials draw the maps, making the process a political battle where each party spends millions of dollars to find ways to reach a competitive advantage in the next decade of elections [1]. The method by which parties gain this advantage is known as "gerrymandering". Through various specialized tactics, politicians are able to achieve district maps in which their party is likely to win far greater districts than are proportionally representative of the state's demographic. This ultimately leads to a myriad of issues including disenfranchisement, voter apathy, less competitive elections, and reduced motivation for representatives to engage with constituents. Even more recognizably, this leads to districts that are in many cases strangely shaped or unnecessarily large. Naturally, there have to be much better, well-defined ways to draw congressional districts based on several criteria. This project aims to employ some of these criteria along with a specification for proportional party representation through an integer linear programming (ILP) optimization model. Furthermore, this project tests and evaluates its findings by

running its model on data from the state of Ohio—a swing state with a history of relentless gerrymandering.

1.1 Criteria and Considerations

There are several criteria which have been defined for the purpose of creating optimal, fair districts. To a certain extent, these are also given as guidelines and/or requirements to states when drawing new lines. The National Conference of State Legislatures gives many of these [2]:

A. Population
    - The population of each district should be roughly the same. For congressional redistricting, the U.S. Constitution requires districts to be virtually equal in population. For state legislative districts, some say 10% deviations between districts are acceptable, however based on many court rulings, states like Colorado require a maximum deviation less than 5% [2].

B. Contiguity
    - All parts of the district should be connected to all other parts via a path within the district.

C. Compactness
    - Districts should be in regular geometric shapes and distances between parts of a constituency should be minimized.

D. Integrity

- Districts should not divide territorial boundaries. Subdivisions like cities, counties, etc. should largely remain intact throughout a single district.
- Districts also should not divide "communities of interest" which are regions by which residents have significant shared interests.

E. Competitiveness
   - Districts should be relatively balanced to avoid "safety" for one political party. This encourages representatives to be as engaging and fair to their constituencies as possible. It also encourages greater voter engagement in elections.

F. Voting Rights Act
   - If there exists a racial minority that has a sufficient population, common political interest, and geographic cohesiveness, under the Voting Rights Act a district plan must not cause that group to have "less opportunity than other members of the electorate" to "elect representatives of their choice" [3]. Essentially, a district plan cannot purposefully or inadvertently dilute the votes of minority voters.

This project prioritizes one additional quality of "good" districts to ensure fairness politically throughout a state:

G. Proportionality
   - The number of districts in a state that one political party is favored to win should correspond to the preference of voters in the state as a whole. This criterion of political fairness is one slowly being adopted nationally and

has now been adopted by Ohio after the passage of Issue 1 regarding legislative redistricting reform [2].

1.2 Compactness Measures

As mentioned in Section 1.1, districts should be made to be compact and should have regular geometric shapes. As humans, it is easy to look at a shape and judge whether we believe it to be sufficiently compact, but especially when using computers to compare district shapes, it is necessary to have a metric that quantifies this level of compactness. Fortunately, there are a great deal of compactness measures in the literature for us to choose from.

1.2.1 Traditional Compactness Measures

Here we highlight some of the most common metrics used in redistricting and case law:

A. Polsby-Popper

- This is a perimeter measure that idealizes the shape of a circle. This measure effectively states that a compact district should have a large area relative to its perimeter. Specifically, it is the ratio of the area of a district to the area of a circle whose circumference is equal to the district's perimeter. The score is computed as follows with higher scores indicating more compact districts:

$$PP = \frac{4\pi {A_D}^2}{P_D}$$

where A_D is the area of the district and P_D is the perimeter of the district [4]. See Figure 1 below demonstrating the relationship between the district and the circle we are idealizing.

Figure 1

A District in Circle whose Circumference Equals District's Perimeter

Note: Figure reprinted from A. Adhia, D. Grozdanov, M. Ramezanzadeh and Z. Fisher, "Measuring Compactness," [Online]. Available: https://fisherzachary.github.io/public/r-output.html. [Accessed 15 April 2021].

- This is a measure of a dispersion, similar to Polsby-Popper in that it idealizes a circle. This measure is the area of a district divided by the area of its smallest circumscribing circle and is computed as follows with higher scores reflecting more compactness:

$$R = \frac{A_D}{A_C}$$

where A_D is the area of the district and A_C is the area of the smallest circle that encloses the district's geometry [5]. See Figure 2 below demonstrating the relationship between the district and the circle being targeted in the ratio.

Figure 2

A District in Smallest Enveloping Circle

Note: Figure reprinted from A. Adhia, D. Grozdanov, M. Ramezanzadeh and Z. Fisher, "Measuring Compactness," [Online]. Available: https://fisherzachary.github.io/public/r-output.html. [Accessed 15 April 2021].

C. Convex Hull

- This measure compares a district area to the area of its minimum enclosing convex polygon and is computed as follows with higher scores reflecting greater compactness:

$$CH = \frac{A_D}{A_P}$$

where A_D is the area of the district and A_P is the area of the minimum convex polygon that encloses the district's geometry [6]. See Figure 3 below demonstrating the relationship between the district and its convex hull.

Figure 3

A District Enclosed by its Convex Hull

Note: Figure reprinted from A. Adhia, D. Grozdanov, M. Ramezanzadeh and Z. Fisher, "Measuring Compactness," [Online]. Available: https://fisherzachary.github.io/public/r-output.html. [Accessed 15 April 2021].

D. Length-Width

- This measure simply uses the ratio of a district's width to its length as found by its minimum enclosing rectangle. This is computed as follows

with the longest edge of the rectangle used as the length such that higher scores indicate greater compactness:

$$LW = \frac{W_R}{L_R}$$

where W_R is the width of the district's bounding rectangle and L_R is the length of the district's bounding rectangle [6]. See below demonstrating the relationship between the district and its minimum enclosing rectangle.

Figure 4

A District Enclosed by its Minimum Bounding Rectangle

Note: Figure reprinted from A. Adhia, D. Grozdanov, M. Ramezanzadeh and Z. Fisher, "Measuring Compactness," [Online]. Available:

https://fisherzachary.github.io/public/r-output.html. [Accessed 15 April 2021].

1.2.2 Limitations of Traditional Compactness Measures

In theory, each of the compactness measures stated above should effectively measure the compactness of most districts. In practice, however, each one has a slew of problems which could cause bad shapes to get good scores or vice versa. Duchin and Tenner explain that this is largely due to the fact that each of these measures are contour-based, utilizing Euclidian based computations of area and perimeter [7]. Duchin and Tenner outline four main reasons these measures can result in inconsistencies:

1. Coastline Effects
    - A district can have a highly inflated perimeter simply because it shares an edge with a natural feature, like a coastline. These jagged edges inflate perimeter aggressively and can lead to poor compactness measures. Polsby-Popper in particular is highly sensitive to this.
2. Resolution Instability
    - The computation of perimeter is heavily impacted by the choice of resolution in a digital map. Higher resolutions will capture finer details of a shape's edge and significantly increase its perimeter. Because resolutions may vary from one map to the next, it can be highly unstable for comparison purposes.
3. Coordinate Dependence

- Naturally, the Earth is spherical but when rendering and performing computations on maps, we must project the sphere onto a plane. There are many choices for such projections, but none can perfectly preserve both area and shape. Thus, computations for area can vary drastically and unpredictably when changing from one projection to the next. This has a particularly high impact on area-based scores like the Reock score. Duchin and Tenner also highlight here that area has no particular relevance to redistricting [7].

4. Empty Space Effects
    - In redistricting, we optimize for equal population, not equal area. Unpopulated regions of a district do not impact election outcomes but can still have a very high impact on these traditional compactness measures. Thus, these measures can vary drastically simply based on how we allocate a mountain or lake to a particular district, which is not relevant to the goals we wish to achieve.

It is also worth noting that the Length-Width measure, which uses neither the area nor the perimeter of the district itself, is highly imperfect. To understand this, simply visualize a district snaking back and forth through a square. Since the length and width will be the same, it will have a perfect compactness score, but a snake-like district such as this is certainly not compact.

1.2.3 Discrete Compactness Measures

Due to the variety of issues related to each of these traditional, Euclidian-based compactness measures, Duchin and Tenner advocate for discrete techniques when computing compactness. While the Supreme Court requires that population be nearly equal between congressional districts, the finest granularity of unit we have available with population totals is the Census block. This means that any district plan generally must be built from some type of discrete units, like Census blocks, in order to properly account for population balance. This lends itself very well to discrete, graph-based techniques when analyzing the compactness of a district.

As proposed by Duchin and Tenner, we can induce a vertex-weighted dual graph of a district plan where the vertices are the basic units of Census geography used. Edges will then exist between vertices that correspond to adjacent units and each vertex will get a weight based on its corresponding unit's population. With this graph, we can define a *discrete area* measure for a single district's corresponding subgraph. This measure is equal to the order of the subgraph, which is the number of units in the subgraph. We can also define a *discrete perimeter* measure equal to the number of units along the subgraph's boundary. This immediately translates into a discrete analog of Polsby-Popper where the discrete area is divided by the square of the discrete perimeter. The same computation can also be done weighted by population for an even better measure. Duchin and Tenner explain how a discretized Polsby-Popper measure like this majorly avoids the pitfalls explained above for Euclidian compactness measures. We effectively eliminate the impact caused by coastlines, varying resolutions, and coordinate systems by

computing our measures solely based on the way we discretely piece together our district's building blocks [7].

1.2.4 Cut Edges

DeFord and Duchin simplify the use of a district plan's dual graph even further in [8]. Here they measure compactness by computing the number of *cut edges* in a district plan. They define a cut edge as any edge in the graph that exists between two units assigned to different districts. A plan with fewer cut edges is more compact as it divides the plan into districts with simpler shapes. They evaluate this measure on Virginia and find it to be highly superior to area-based measures [8]. Thus, this is the measure we ultimately use to constrain compactness in our model due to its simplicity and consistency.

1.3 Partisan Fairness Measures

Section 1.1 describes one partisan fairness metric which is a key focal point for the model described in this project: proportionality. This is a fairly straightforward measure of partisan fairness where the proportion of districts that a given party wins should be proportional to that party's voters statewide. However, there are several other measures for partisan fairness. We briefly describe two alternative measures here: efficiency gap and partisan symmetry.

1.3.1 Efficiency Gap

Partisan gerrymandering effectively seeks to force a particular party to waste votes by packing that party's voters in few districts and cracking the favored party's voters into a majority share of as many districts as possible. The efficiency gap, as

proposed in [9], seeks to effectively compare how efficient a district plan is for each party by comparing the proportion of wasted votes for each party. Votes are considered to be wasted when they don't help a party to win more districts. Any vote for a losing party is considered to be wasted and any vote for a winning party beyond the simple majority is considered to be wasted. These wasted votes are summed together for each party separately and the difference between wasted votes divided by the total number of voters statewide yields the efficiency gap. The closer this gap is to 0%, the fairer the district plan. The authors of the efficiency gap argue that 8% is a significant threshold for when a district plan becomes unacceptable in terms of partisan gerrymandering [9].

While this metric has significant merit and has gained traction in the Supreme Court as an acceptable indicator of partisan gerrymandering, it has its flaws. Bernstein and Duchin show that under the assumption of equal voter turnout among districts, the efficiency gap effectively boils down to be the statewide vote lean favoring the winning party minus half of the statewide seat lean favoring that party:

$$EG \approx \frac{T_A - T_B}{T} - \frac{1}{2} * \frac{S_A - S_B}{S}$$

Here, T is the number of voters and S is the number of seats with indices A and B used to differentiate votes and seats by party. The biggest problem that this result demonstrates is that for the efficiency gap to be 0, the seat lean should be twice the vote lean. This directly penalizes proportionality and can cause a district with perfect proportionality to be above the 8% threshold for efficiency gap [10].

1.3.2 Partisan Symmetry

Another way to define partisan fairness in a district plan is by using the partisan symmetry standard. This standard requires that each party receive the same proportion of legislative seats as the other would have given the same share of the statewide vote [11]. For example, if Republicans win 70% of the seats with 55% of the statewide vote, then this is considered fair if Democrats would also win 70% of the seats had they won 55% of the statewide vote.

This standard can be measured by a variety of scores with two of the most common being the mean-median metric and the partisan bias metric. Each of these are based on the core principle that if a party wins half the votes, it should win half the seats. The mean-median metric measures how many votes short of half that a party can win while still winning half the seats. Partisan bias, however, measures how many seats beyond half the seats a party can win while winning half the statewide share of votes. The closer these scores are to 0, the fairer the map [12].

However, like efficiency gap, these metrics and the way they are computed have been found to have serious issues. In particular, it is shown in [12] that these metrics do not necessarily constrain extreme partisan gerrymandering and can have unforeseen consequences on partisan outcomes. In some cases, it is shown that these metrics can even fail to identify which party is at an advantage in a district plan [12].

1.4 Alternative Computational Approaches to Redistricting

The mathematical model proposed in this paper is one example of using an algorithm for redistricting. To our knowledge, it is one of the only integer linear programming applications for redistricting that constrains proportionality. It also defines

and proves the correctness of a completely novel method for constraining contiguity. However, there have been many other proposed systems of algorithmic redistricting that use various other techniques that don't rely on integer programming. A fairly comprehensive summary of redistricting algorithms is given in [13] and here we summarize some of the main categories.

1.4.1 Enumeration

This is essentially a brute force approach to redistricting in which all possible valid plans are generated. While this has been proposed several times in the literature, it is impossible to use in practice except in the very smallest of instances. With redistricting, as the problem gets larger with more units, the number of valid district plans increases exponentially. For example, even in the fairly simple case of building 4 districts out of Iowa's 99 counties, there would be 4^{99} ways to assign districts to counties, which is on the order of 10^{59}. Of these, there is estimated to be 10^{24} valid solutions and well over 99% of these would be highly non-compact [13]. Thus, enumeration is impractical.

1.4.2 Random Assignment and Rejection Sampling

This is a very simple approach to redistricting that simply assigns each unit a district label. Naturally, generating a plan randomly this way is unlikely to even achieve contiguity, so rejection sampling is used to discard invalid plans. The algorithm continues to sample plans until a valid one is found. As you can imagine, with so many possible district assignments, this is shown to not only be inefficient, but simply ineffective.

1.4.3 Flood Fill

This class of algorithms uses random seed units from which districts are grown by joining adjacent units until a target population is reached. There are several variations on this algorithm, but one of the main challenges is that it can get stuck. Districts can be grown in a way that does not allow for balanced, contiguous districts so these must be rejected, and the algorithm restarted.

1.4.4 Flip-Step

Instead of generating plans from scratch, some algorithms start from a valid district plan and use it to generate other plans. One of these algorithms is flip-step which randomly samples a unit on one of the districts' boundaries. This unit is then *flipped*, which means it is reassigned to one of the districts it neighbors—so long as this flip does not make the plan invalid. This process is repeated up to as many as millions or billions of times to create a new plan with little resemblance to the original.

1.4.5 Recombination

Recombination is another algorithm that uses a valid plan as input. This algorithm takes steps toward a new plan by merging adjacent districts and repartitioning them into new ones. Recombination has been used in particular to generate thousands of plans for the purpose of representing the distribution of possible plans that meet traditional redistricting criteria. This can be used to represent the typical properties that one can expect of a state's plan simply based on its geography and Census data. This provides a baseline for comparing some other proposed plan to help argue whether some of its characteristics are typical or whether the plan is an outlier—indicating evidence of gerrymandering [14].

1.4.6 Random Walk Metaheuristics

There are also many variations of the random walk algorithms flip-step and recombination described in 1.4.4 and 1.4.5. These employ various metaheuristics to take steps not just toward other random plans in the state space, but toward more optimal ones. This is done by defining some type of scoring method for a district plan based on any set of criteria. During any recombination or flip step, instead of choosing the step randomly, each possible choice is evaluated against this score. One metaheuristic, hill climbing, always chooses the most optimal step to take until there exist no steps that can improve the plan's score. However, this heuristic is likely to lead toward a local optimum, so it is typically applied to a variety of starting plans and the best scoring plan among all the runs is chosen. Other metaheuristics applied to random walks include a simulated annealing variant of hill climbing, Tabu search, and various evolutionary algorithms.

1.4.7 Voroni Approaches

Other algorithms ignore basic building block units in favor of a purely geometric approach that prioritizes compactness. Some such algorithms use Voroni diagrams where points are chosen on the map to be district hubs. Every other part of the state is then assigned to a district based on whichever hub it is closest too. One of the biggest challenges with this approach is in deciding the optimal placement of the district hubs. This also does not take population balance into account, so a variation on this method uses power diagrams where each hub has an associated weight. With these diagrams, balanced and compact districts can be achieved but they suffer from the fact that Census

units are not respected. This means that it can be difficult to properly assign population to each district and definitively declare that they are balanced.

1.5 Preview

In Chapter 2 we describe the general integer linear programming-based mathematical model proposed to help draw optimal districts. Here we provide the input data, decision variables, objective function, and constraints used to make in the model as well as provide arguments and proofs along the way proving its validity. In Chapter 3 we apply the mathematical model to Ohio and discuss our process selecting and preparing the input data. We also provide a breakdown of computational results from two datasets and compare our model's results to that of the state's current district plan. Finally, in Chapter 4 we offer several future directions to expand on this project to better meet each of the redistricting criteria and summarize our model and its potential impact on redistricting.

Chapter 2: Mathematical Model

In an effort to achieve optimized districts with so many competing district qualities, we propose an integer linear programming (ILP) optimization model which can be implemented to formalize and optimize this process. The mathematical model is implemented and tested using the algebraic modeling language AMPL. The complete AMPL code used to implement and test this model can be found in Appendix A. In this chapter we give the data, variables, objective function, and functional constraints used to build the model. We also give arguments proving the validity of the model.

2.1 Input Data: Sets and Parameters

Let $\boldsymbol{Units}$ be a set of $\boldsymbol{N}$ basic geographic units in Ohio. Let $\boldsymbol{Districts}$ be a set of $\boldsymbol{M}$ electoral districts. A unique subset of units should be assigned to each district.

2.1.1 Defining Restricted Possibilities

While we could allow the model free range to assign any possible unit to any possible district, this would require $N * M$ decision variables, which would be an excessive waste of time and memory as it is impractical to consider units on opposite sides of the state for the same district. To alleviate this complexity, we define a parameter for whether a particular unit is allowed to be included in a given district thus eliminating any possible district assignments that would never make sense. Thus, we define the following binary parameter:

$$\text{Let } p_{ij} = \begin{cases} 1, \text{if unit } i \text{ can be included in district } j \\ 0, \text{otherwise} \end{cases}$$

$$\forall\, i \in Units, j \in Districts$$

2.1.2 Defining Unit Relationships

In order to implement constraints that guarantee connectivity and compactness within a district, we must define the relative distance between any pair of units. The distance defined by this model is the length of the shortest path between two units in terms of the number of adjacent units that must be traveled through to reach the opposing one. If units are adjacent, this distance is 1. More information on how these distances can be computed based on adjacency data for the units in a state can be found in Appendix B. We also define a tuning parameter here that expresses the maximum allowable distance

between any pair of units in a chosen district. Thus, we define the following two parameters:

Let $dist_{ik} \ \forall \ i,k \ \in Units$

be the minimum number of edges that must be traversed to reach k from i

Let $maxDist$ be the maximum allowable distance between any two units in a district

2.1.3 Expressing Demographics

One of the primary features of this ILP model is that it chooses districts such that the number of districts leaning toward any particular party are proportional to the political leanings of the state. To achieve this, we need a number of pieces of demographic political data regarding the number of Democrats, Republicans, and total voters in each unit. We also define a value $LARGE$ that is initialized to the total number of voters in the state. This value acts as a maximum, or relative infinity value, in the context of the model for later logical constraints. Thus, we define:

Let dem_i be the number of Democrats in unit i, $\quad \forall \ i \in Units$

Let rep_i be the number of Republicans in unit i, $\quad \forall \ i \in Units$

Let $vot_i = d_i + r_i, \quad \forall \ i \in Units$

Let $LARGE = \sum_{i \in Units} v_i$

Next, the model defines two important target values: the fair number of Democrat districts and the ideal population of each district. The targeted fair number of Democratic districts, $targDem$, is defined by the proportion of Democratic voters multiplied by the

number of districts. It is important to note that we choose Democrat here arbitrarily. Since this model assumes a two-party system, the fair number of Republican districts is uniquely determined by *targDem* and the result would be the same if the model were formulated in terms of the target number of Republican districts. Thus, *targDem* is defined as follows:

$$\text{Let } targDem = round\left(\frac{\sum_{i \in Units} dem_i}{\sum_{i \in Units} vot_i} * M\right)$$

The ideal population for a district, $targPop$, is defined as the number of statewide voters divided by the number of districts:

$$\text{Let } targPop = round\left(\frac{\sum_{i \in Units} vot_i}{M}\right)$$

Finally, we define a tuning parameter, ε, for the allowable error of the actual number of Democrat districts from the ideal number of these districts in theory. We also define a hyperparameter for the maximum number of cut edges allowed in the district plan which is explained further in (2.4.5.2). Thus, we define:

$$\text{Let } 0 \leq \varepsilon \leq targDem \ (\varepsilon \in \mathbb{Z}) \text{ be the allowable error of the actual number of Democrat districts from the target}$$

$$\text{Let } maxCut \text{ be the maximum number of cut edges allowed in the model's produced district plan}$$

2.2 Decision Variables

We now define the variables needed by the model. The first of these is our primary decision variable, which is a binary variable used to decide whether a particular unit is included in a particular district. This variable, x_{ij}, is defined:

$$\text{Let } x_{ij} = \begin{cases} 1, \text{if unit } i \text{ is chosen for district } j \\ 0, \text{otherwise} \end{cases}$$

$$\forall\, i \in Units, j \in Districts \text{ s.t. } p_{ij} = 1$$

The next variable, z, is an auxiliary variable needed for the objective function as this variable will be minimized in (2.3). This is defined as follows:

$$\text{Let } z \text{ be the largest difference (by absolute value) from}$$

$$\text{the target population}$$

Each of the remaining variables are auxiliary variables needed for writing corresponding constraints. The first of these, y_j, is used in (2.4.4) to help constrain the political distribution of the district plan:

$$\text{Let } y_j = \begin{cases} 1, \text{if more Democrat voters in } j \text{ than Republican voters} \\ 0, \text{otherwise} \end{cases}$$

$$\forall\, j \in Districts$$

The final variables are all used in (2.4.5.2) to help constraint the model to ensure compact districts by limiting the number of cut edges. These are defined as follows:

$$\text{Let } s_{ik} = \begin{cases} 1, \text{if adjacent units } i \text{ and k are in the same district} \\ 0, \text{otherwise} \end{cases}$$

$$\forall\, i, k \in Units, \text{s.t. } dist_{ik} = 1$$

$$\text{Let } w_{ikj} \text{ be an auxillary binary variable}$$

$$\forall\, i, k \in Units, j \in Districts$$

$$\text{s.t. } dist_{ik} = 1, p_{ij} = 1, and\ p_{kj} = 1$$

2.3 Objective Function

Currently, the strictest requirement during redistricting is that districts remain near equal. This requirement is one strictly upheld by the courts under constitutional law in the 14th amendment [2]. For this reason, the model given here will seek to produce a solution with the smallest possible maximum deviation from a district's target population. This is done by minimizing the decision variable z and requiring that variable to be the largest population deviation from the target population of all districts. This objective function is as follows:

$$\min z = \max_{j \in Districts} \left| \left(\sum_{i \in Units,\ s.t.\ p_{ij}=1} x_{ij} * v_i \right) - targPop \right|$$

The way this objective function is linearized for the mathematical model is explained further in (2.4.1).

2.4 Functional Constraints

With everything in place to generate optimal districts in terms of even population, we can begin constraining the model to pick a map that is valid, contiguous, fair, and compact.

2.4.1 Constraining Objective Function Decision Variable z

First and foremost, we must constrain the decision variable z to be the value of the objective function. Unfortunately, the objective function given for z in (2.3) is nonlinear. In order to linearize it, we require z to be equal to the objective function

through two constraints, one for positive deviations from $targPop$, (C1), and one for negative deviations (C2).

$$z \geq \left(\sum_{i \in Units:\, p_{ij}=1} x_{ij} * v_i \right) - targPop, \qquad \forall\, j \in Districts \tag{C1}$$

$$z \geq\ targPop - \left(\sum_{i \in Units:\, p_{ij}=1} x_{ij} * v_i \right), \forall\, j \in Districts \tag{C2}$$

Notice that through these two constraints, we require z to be greater than the population deviation by absolute value *for all of the districts*. This in turn has the same result as making z larger than the maximum deviation, allowing us to then minimize z in the objective function to achieve our desired result.

2.4.2 Constraining Unit-District Relationship

Next, we must force the model to use the decision variable x properly such that each unit is assigned to exactly one district. Using (C3), we are able to force this. Note that we don't explicitly require the converse, that each district is assigned at least one unit. This is because an empty district would produce a very large population deviation equal to the target population. Thus, this constraint is captured by the objective which optimally evens out the population distribution among all districts.

$$\sum_{j \in Districts:\, p_{ij}=1} x_{ij} = 1, \qquad \forall\, i \in Units \tag{C3}$$

2.4.3 Constraining Contiguity

It is now important to ensure that any generated districts are made up of units that are contiguous. This means there must always exist a path between two units in the same

district such that all units along that path are also in the same district. Constraint (C4), below, accomplishes this. However, writing a linear constraint that guarantees contiguity is non-trivial. Thus, we provide both an intuitive explanation for this constraint's functionality as well as a formal proof guaranteeing each district's contiguity.

$$\sum_{\substack{i \in Units: \\ (dist_{ik}=1 \text{ and } dist_{il}=d-1) \\ or\ (dist_{ik}=d-1 \text{ and } dist_{il}=1)}} x_{ij} \geq x_{kj} + x_{lj} - 1, \quad \forall\, k, l \in Units, j \in Districts, \quad s.t.\ d = dist_{kl} \in [2, maxDist], p_{kj} = 1, p_{lj} = 1 \tag{C4}$$

2.4.3.1 Intuition for the Contiguity Constraint's Functionality

What constraint (C4) says in layman's terms is that for every pair of units included in the same district and a distance $d \geq 2$ apart, we require that one of that pair's ***bridge units*** also be included in the district. A bridge unit is a term we coined and is defined as follows:

> **Definition.** For any two units that are a distance $d \geq 2$ apart (i.e., there are at least two edges separating them), a ***bridge unit*** is defined to be a unit that is adjacent to one of them and exactly a distance of $d - 1$ from the other.

Consider the graphic in Figure 5. Let every enclosed space be a unit in the state. Consider k and l to have already been chosen for the same district. Let $dist_k$ be the distance of a given unit from k and $dist_l$ be the distance of a given unit from l. Notice that k and l are a distance of 5 from one another. Let those units highlighted light blue be

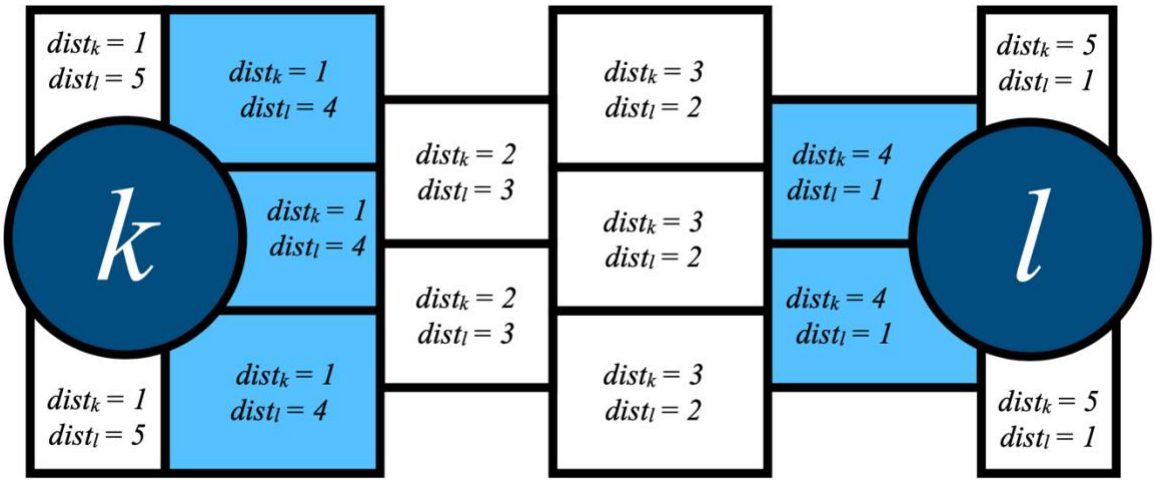
the set of bridge units (as they are a distance of 1 from either k or l and a distance of $5-1=4$ from the other).

Intuitively, via the constraint above, we require that at least one of these light blue bridge units also be included in the district. Notice that the summation selects for candidate bridge units. Only when both units *k* and *l* are included in the same district will the right-hand side be equal to 1. When this is the case, the constraint will force the model to include at least one of the candidate bridge units highlighted in Figure 5.

Since these bridge units bring the connectedness of the disjoint subgraphs containing k and l closer together and the included bridge unit will then require its own bridge unit to be included in the district, from a recursive perspective, bridge units will continue to be added until these subgraphs are connected via some path.

Figure 5

Diagram Demonstrating Bridge Unit Definition



2.4.3.2 Contiguity Proof

We now provide a formal proof that constraint (C4) achieves connectedness. First, recall the definition of a connected graph:

> **Definition.** A graph *G* is ***connected*** if and only if for any two nodes $x, y \in G$, $\exists$ a sequence of nodes $b_1, b_2, \ldots, b_n$ s.t. $dist_{xb_1} = 1$, $dist_{yb_n} = 1$, and $\forall\ i \in [2, n],\ dist_{b_{i-1}b_i} = 1$

The main connectedness result is given in the following theorem:

> **Theorem.** All resulting districts from the redistricting model will always contain a set of units U which can be expressed as a connected graph where nodes are the units in a district and edges exist between adjacent units.
>
> **Proof.** We must show that for any pair of units in U corresponding to its district j, that they are guaranteed to be connected by some subset (or component) of nodes from the district graph. We will show it *by induction* over possible distances d in the range $[1, maxDist]$ for a pair of units in the district.
>
> *Base Cases:*
>
> $d = 1$ (Trivial Case):
>
> Consider any two nodes k, l in a district graph to be distance 1 apart. If this is the case k and l are connected by default (by the definition of connectedness).
>
> $d = 2$:

Consider any two nodes k, l in a district graph to be distance 2 apart. By the (C4) constraint, some node i in the set of

$$\left\{ \begin{array}{c} i \in Units, \;\; s.t. \\ (dist_{ik} = 1 \; and \; dist_{il} = 2 - 1) \\ or \; (dist_{ik} = 2 - 1 \; and \; dist_{il} = 1) \end{array} \right\}$$

must be included in the district graph. By the definition of this set in (C4), i must be a distance of 1 from both k and l. Thus, by the definition of connectedness, k and l are connected via i.

Inductive Step:

Thus, by proving the above base cases we can assume the following ***inductive hypothesis***: Assume that any two nodes (units) in the district graph distance d from one another are connected for any d s.t. $0 < d < maxDist$.

We must now show that any two nodes k, l in the district graph, such that k, l are distance $m = d + 1$ apart, are connected.

By the constraint (C4), some node i such that i is adjacent to k and distance $m - 1$ from l or such that i is adjacent to l and distance $m - 1$ from k must be included in the district graph.

By the inductive hypothesis, we have that for any two nodes in the district graph of distance less than m are connected. Thus, in the first case for i it must be the case that i is adjacent to k and connected to l by the inductive hypothesis. In the second case for i it must be that i is adjacent

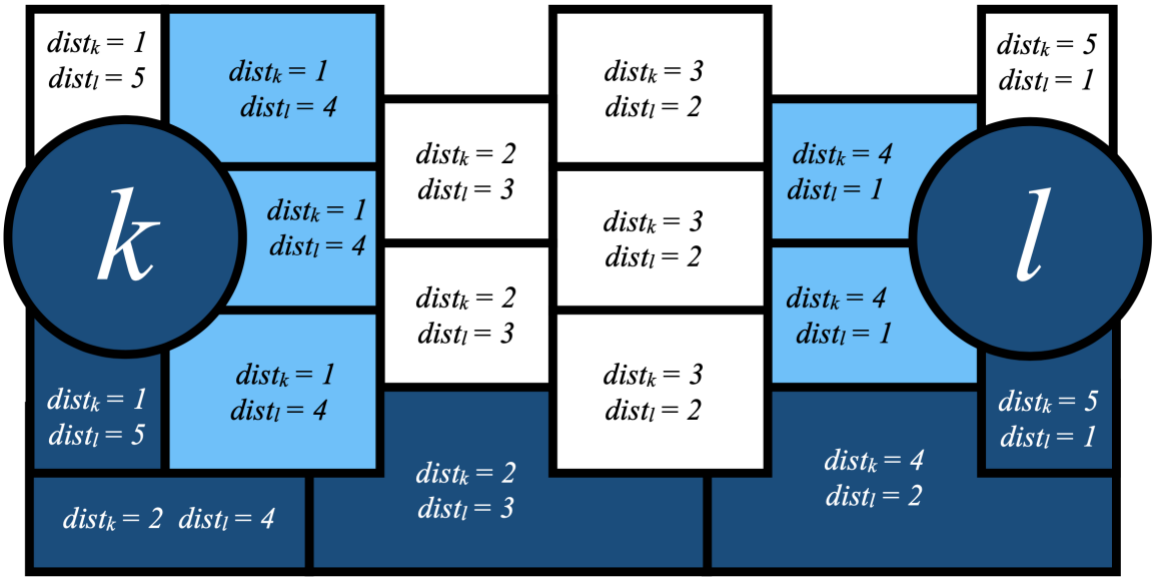
to l and connected to k by the inductive hypothesis. In either case, by the definition of connectivity, k is connected to l via the included node i. Qed.

2.4.3.3 Additional Restrictiveness

It is important to note that while this constraint, (C4), guarantees connected districts as shown in the proof, having this constraint prohibits a few possible ensembles of districts. Consider a modification to Figure 5 as shown in Figure 6.

Figure 6

Diagram Demonstrating Contiguity Constraint Restrictiveness



Let all the dark blue units be included in the same district as k and l. Notice that the inclusion of these units would make the dark blue district fully contiguous. However, it is clear by inspection that none of these units fit the criteria to be considered a bridge

unit for the pair of units k and l. Thus, their inclusion will not be sufficient for our model to consider k and l to be connected and it will require additional bridge units (one of the light blue shaded units) to be included as a result. Thus, a resulting district including only those units highlighted dark blue as above would not be possible under the constraints given in this model. This creates an additional, unintended restrictive property by including the contiguity constraint. It is important to realize this side effect and the cost of having it, however it can largely be seen as a bonus benefit to the connectivity constraint. It is evident that districts like the one considered here are visibly not compact and would tend to produce enclaves. These are not desirable qualities anyway so omitting these possibilities has little cost to the targeted solution space of the model.

2.4.4 Constraining Political Distribution

Next, we constrain the model to produce districts that provide partisan fairness through proportionality. This means the model will be forced to choose districts such that the proportion of districts that favor one party will be close to the proportion of that party's voters statewide. In order to implement such a constraint, we require a constrained variable, y_j, that indicates whether a particular district has more Democrats than Republicans in it.

$$LARGE * y_j \geq \sum_{i \in Units:\, p_{ij}=1} \left(x_{ij} * dem_i - x_{ij} * rep_i\right), \quad \forall\, j \in Districts \tag{C5}$$

This binary variable is set to 1 by constraint (C5), above, if this is the case. This works because the right-hand side will be positive when there are more Democrats, thus forcing y_j to be nonzero. Note that $LARGE$ will always satisfy this inequality since it is

initialized to the total number of voters in the state and must be greater than the difference between the parties' voters.

In the case where there are more Republicans, (C5) provides no constraint to y_j. Thus, we also utilize (C6), below, which in the case of a district having more Republicans, is positive on the right-hand side. This works similarly and forces $(1 - y_j)$ to be nonzero, thus forcing y_j to be 0.

$$LARGE * (1 - y_j) \geq \sum_{i \in Units:\, p_{ij}=1} (x_{ij} * rep_i - x_{ij} * dem_i), \quad \forall\, j \in Districts \tag{C6}$$

Now that the model can keep track of the number of districts that favor each party, we can constrain this number using our predefined target, $targDem$. In order to allow some leeway on this target number, we use the hyperparameter ε. Through (C7) and (C8), we force the model to choose districts such that the number of them that lean Democrat is equal to $targDem \pm \varepsilon$.

$$\sum_{j \in Districts} y_j \leq targDem + \varepsilon, \qquad \forall\, j \in Districts \tag{C7}$$

$$\sum_{j \in Districts} y_j \geq targDem - \varepsilon, \qquad \forall\, j \in Districts \tag{C8}$$

2.4.5 Constraining Compactness

We want to make sure our model captures another key criterion of good districts—compactness. Typically, in districts that are more spread out, voters are unlikely to share as strong of a geographic and cultural identity, and therefore might have fewer shared concerns. This makes it harder for effective representation and in addition,

non-compact district can be a tell-tale sign of a gerrymandered map. Thus, we provide two methods for constraining compactness in order to ensure as compact of districts as possible.

2.4.5.1 Maximum Distance

First, we can constrain for compact districts by setting a maximum distance between any two units in a particular district. We create this constraint by considering every pair of units greater than the maximum distance and not allowing them to be in the same district.

$$x_{ij} + x_{kj} \leq 1, \qquad \forall\, i,k \in Units, j \in Districts\ s.t.\ p_{ij} = 1, \\ p_{kj} = 1, and\ dist_{ik} > maxDist \tag{C9}$$

2.4.5.2 Limiting Cut Edges

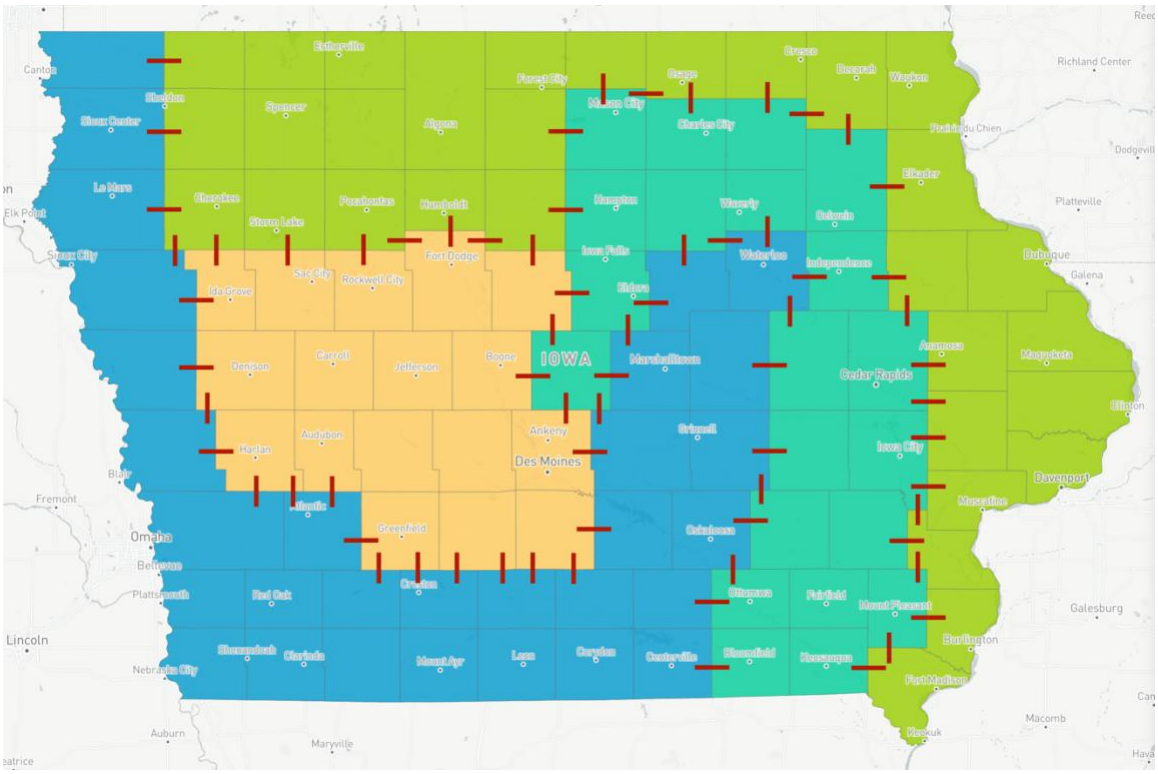
A more sophisticated measure we can utilize to ensure we have a compact district is known as the number of ***cut edges***.

> **Definition.** A ***cut edge*** is an edge between two adjacent units in a state that have been separated into different districts.

District plans with fewer cut edges are more compact. As an illustrative example, see in Map 1 a district plan of Iowa, which is used here since Iowa districts must be built from large county-level units, making for a simple, practical example. Cut edges are marked by red dashes along unit boundaries. Notice that the districts in this plan are not compact at all and sprawl throughout the state. This plan has 68 cut edges. Then, see in Map 2 a district plan of Iowa with highly compact districts. Notice that this plan has far fewer cut edges with only 30 as opposed to 68.

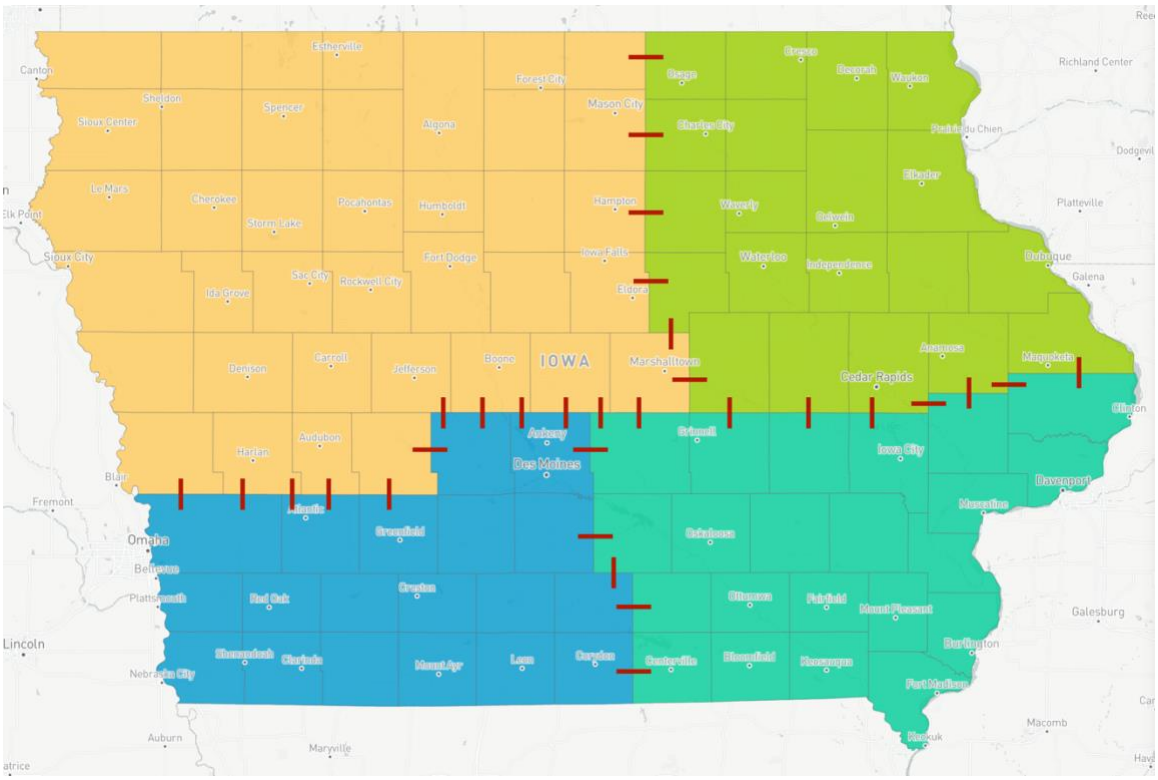
Map 1

Congressional District Plan of Iowa with Non-Compact Districts and 68 Cut Edges



Map 2

Congressional District Plan of Iowa with Highly Compact Districts and 30 Cut Edges



Thus, we can utilize this measure to constrain the model to only choose a plan up to a certain number of cut edges. In order to do this, we require a few auxiliary variables as defined in (2.2). The first of these, s_{ik}, is used to determine whether neighboring units are in the same district and is equal to 1 if this is the case or 0 if not. The following three constraints ensure this using an additional auxiliary variable, w_{ikj}:

$$s_{ik} \geq c_{ij} + c_{kj} - 1, \quad \forall\, i,k \in Units, j \in Districts\ s.t.\ dist_{ik} = 1, \quad p_{ij} = 1, and\ p_{kj} = 1 \qquad \text{(C10)}$$

$$s_{ik} \leq 0.5 * \left(c_{ij} + c_{kj}\right) + \left(1 - w_{ikj}\right),$$

$$\forall\, i,k \in Units, j \in Districts\ s.t.\ dist_{ik} = 1, \qquad \text{(C11)}$$

$$p_{ij} = 1, and\ p_{kj} = 1$$

$$\sum_{\substack{j \in Districts: \\ p_{ik}=1\ and\ p_{kj}=1}} w_{ikj} \geq 1, \qquad \forall\, i,k \in Units, s.t.\ dist_{ik} = 1 \qquad \text{(C12)}$$

Notice that the first constraint, (C10), will ensure s_{ik} is 1 any time two neighboring units are in the same district, as the right-hand side will be 1. If the units are not included in the same district, (C10) places no restriction on s_{ik}.

The second constraint, (C11), in conjunction with (C12) ensure that s_{ik} is 0 if i and k are not included in the same district, without placing any restriction on s_{ik} if they are in the same district. To understand this, first consider the case where i and k are not included in the same district. This means given any particular district, j, $0.5 * \left(c_{ij} + c_{kj}\right)$ can only be at most 0.5. Constraint (C12) ensures that for at least one district j, $w_{ikj} \geq 1$ and therefore $1 - w_{ikj} \leq 0$. Summing these terms on the right-hand side of (C11) means that for at least one district j, this constraint will force that $s_{ik} \leq 0.5$ and since s_{ik} is a binary variable, it must be 0.

Now consider the case where i and k are included in the same district. Constraint (C10) will force s_{ik} to be 1, but we must make sure that (C11) provides no contradictory restriction. This is where the slack variable w_{ikj} serves its main purpose. When the constraint considers the districts i and k are not included within, $0.5 * \left(c_{ij} + c_{kj}\right) = 0$, but w_{ikj} can be 0 as well still making the right-hand side equal to 1, thus placing no

restriction. When the constraint considers the district which i and k are both included in, $0.5 * (c_{ij} + c_{kj})$ will be 1, placing no restriction on s_{ik}, and w_{ikj} can be 1 as well in order to still satisfy (C12).

Now that we have a variable indicating whether two neighboring units are in the same district, we can write a constraint that sums up and limits the number of cut edges:

$$\frac{1}{2} * \left(\sum_{\substack{i,k \in Units: \\ dist_{ik}=1}} (1 - s_{ik}) \right) \leq maxCut \qquad \text{(C13)}$$

First, notice that the right-hand side of (C13) sums together all of the s_{ik} variables, but first subtracts each from 1. The effect of this transformation is to make s_{ik} 1 if it was originally 0 and to make s_{ik} 0 if it was originally 1. Without this transformation, we would get the sum of all the *uncut edges* in a district plan, where both units are in the same district. By applying this linear transformation and then summing the variables, we in effect compute twice the number of *cut edges*. This is because the s_{ik} variables resemble a symmetric matrix where s_{ik} is the same variable as s_{ki}. Thus, we can multiply this summation by ½ and after computing this number of cut edges, we can limit that value by the parameter $maxCut$.

2.5 Summary of Model

Here we provide a summary of the model's objective and constraints in one place.

Objective

min z

subject to:

$$z \geq \left(\sum_{i \in Units:\, p_{ij}=1} x_{ij} * v_i \right) - targPop, \qquad \forall\, j \in Districts \tag{C1}$$

$$z \geq\ targPop - \left(\sum_{i \in Units:\, p_{ij}=1} x_{ij} * v_i \right), \forall\, j \in Districts \tag{C2}$$

Unit-District Relationship

$$\sum_{j \in Districts:\, p_{ij}=1} x_{ij} = 1, \qquad \forall\, i \in Units \tag{C3}$$

Contiguity

$$\sum_{\substack{i \in Units: \\ (dist_{ik}=1\ and\ dist_{il}=d-1) \\ or\ (dist_{ik}=d-1\ and\ dist_{il}=1)}} x_{ij} \geq x_{kj} + x_{lj} - 1, \tag{C4}$$

$$\forall\, k, l \in Units, j \in Districts,$$

$$s.t.\ d = dist_{kl} \in [2, maxDist], p_{kj} = 1, p_{lj} = 1$$

Political Distribution

$$LARGE * y_j \geq \sum_{i \in Units:\, p_{ij}=1} \left(x_{ij} * dem_i - x_{ij} * rep_i \right), \tag{C5}$$

$$\forall\, j \in Districts$$

$$LARGE * \left(1 - y_j\right) \geq \sum_{i \in Units:\, p_{ij}=1} \left(x_{ij} * rep_i - x_{ij} * dem_i \right), \tag{C6}$$

$$\forall\, j \in Districts$$

$$\sum_{j \in Districts} y_j \leq targDem + \varepsilon, \qquad \forall\, j \in Districts \tag{C7}$$

$$\sum_{j \in Districts} y_j \geq targDem - \varepsilon, \qquad \forall\, j \in Districts \tag{C8}$$

Compactness—Maximum Distance

$$x_{ij} + x_{kj} \leq 1, \qquad \forall\, i,k \in Units, j \in Districts\ s.t. p_{ij} = 1, \\ p_{kj} = 1, and\ dist_{ik} > maxDist \tag{C9}$$

Compactness—Cut Edges

$$s_{ik} \geq c_{ij} + c_{kj} - 1, \\ \forall\, i,k \in Units, j \in Districts\ s.t.\ dist_{ik} = 1, \\ p_{ij} = 1, and\ p_{kj} = 1 \tag{C10}$$

$$s_{ik} \leq 0.5 * \left(c_{ij} + c_{kj}\right) + \left(1 - w_{ikj}\right), \\ \forall\, i,k \in Units, j \in Districts\ s.t.\ dist_{ik} = 1, \\ p_{ij} = 1, and\ p_{kj} = 1 \tag{C11}$$

$$\sum_{\substack{j \in Districts: \\ p_{ik}=1\ and\ p_{kj}=1}} w_{ikj} \geq 1, \qquad \forall\, i,k \in Units, s.t.\ dist_{ik} = 1 \tag{C12}$$

$$\sum_{\substack{i,k \in Units: \\ dist_{ik}=1}} (1 - s_{ik}) \leq maxCut \tag{C13}$$

Chapter 3: Computational Results

Here we provide a summary of our data preparation and experimental results when validating the model on the state of Ohio. Additional results alongside the code used in each section will be made available on the project's public repository at https://github.com/benjamincarman/repairing-redistricting.

3.1 Data Preparation

For our proposed model to function, we require a set of base-level geographic units from which the model will build districts by grouping these units together. Furthermore, we need data attached to these geographic units in order to use objectives and constraints related to population balance, partisan balance, connectedness, and compactness. In particular, this data includes population totals, vote totals in past elections, and the relative "distance" between any two units as determined by the minimum number of edges one must travel through to reach the other.

3.1.1 Granularity of units

We focus on testing and validating our model using the heavily gerrymandered state of Ohio. In order to begin choosing and collecting base units for Ohio one primary concern is that we require election data attached to the geographic units. In Ohio, the smallest geographic unit at which election totals are reported are known as precincts. This means the smallest granularity of units we can provide the model would be a precinct map of the state of Ohio. Unfortunately, Ohio has 8,882 precincts and in testing, these units were too computationally intensive given our limited computing power and time. This means we had to determine and collect some other unit options with decreased granularity in order to reduce the computational complexity. First, the following options were all prepared and tested:

- Precincts
    - Ohio does not publish or maintain a digitized precinct map of the state. In order to build maps from basic precincts, we use data and maps collected

and prepared by the Metric Gerrymandering and Geometry Group (MGGG) based out of Tufts University [15]. This dataset contains 8,882 units and was too computationally intensive to be used but was used to build all of other datasets explained below.

- Places
    - These units were manually created to provide fine granularity but with only 1,204 units. These were generated using the set of Census designated places as downloaded from NHGIS [16]. Census places are simply points in Ohio that correspond to local cities, towns, and villages. Precincts were then grouped together based on the closest Census designated place in the same county to make a map of small geographic areas centered around Census places.
- Counties with three densest divided into places
    - Since precincts nest perfectly into counties and preserving counties is a goal (integrity) during redistricting anyway, we considered using a county-level map. Still, at the very least the three most populous counties in Ohio—Cuyahoga, Franklin, and Hamilton—need to be split into smaller units as they are generally too dense to entirely fit within one district. For these counties, we split them into the place-level units as described above. This left us with 253 units in total.

Through the creation of each of these options from precents to places to counties, the granularity of the base units decreases, which also decreases the computational

complexity of the model by reducing the number of variables. On the other hand, by using smaller units (higher granularity) we provide the model with more flexibility to find the most optimal result. Given our limited computational resources and time, we still discovered that each of these first three options for base units were too computationally complex. We needed to reduce the number of units even further and settled on using the county-level map of Ohio where some of the largest counties were split into sub-units units as seen in Map 3 and Map 4 below. These datasets ultimately used for validating our model are described next.

3.1.1.1 Dataset 1

This dataset is built off of the counties dataset where the 3 densest counties were split into place units. We reduced the granularity of this dataset further by manually grouping the place-level units in the three densest counties into five separate units of roughly similar area. This left 100 units and they can be seen in Map 3.

Map 3

County map of Ohio with 3 most populous counties divided manually

3.1.1.2 Dataset 2

This dataset is similar to Dataset 1 as most units are simply Ohio's counties. However, in this dataset in order to make population as equal as possible among units, the nine most populous counties were divided into sub-units: Cuyahoga, Franklin, Hamilton, Summit, Stark, Lucas, Montgomery, Butler, and Lorain. While these counties

were divided by hand, extra care was taken to make sure each of the sub-units were as equal in population as possible. This left 105 units and they can be seen in Map 4.

Map 4

County map of Ohio with 9 most populous counties divided manually into even sub-units

3.1.2 Supplying model parameters

Given a map of units, vote and population totals were aggregated using the Geopandas Python library from precinct totals provided by data from MGGG [15]. For

the purposes of these tests, the number of Democrats and Republicans in each unit were simply determined based on the 2016 US presidential election vote totals. A more methodical approach could be used in production for determining an accurate representation of a unit's political leanings.

The parameter p_{ij}, given in 2.1.1, for whether a unit was able to be included in a given district was defined methodically such that a given unit could be included in its current district (from the 2012-2022 map) or any district adjacent to its current district. This still allowed for a great amount of flexibility in the model's options while also ensuring many unnecessary decision variables were removed, thus improving the model's complexity.

Finally, data was included for the distances between each pair of units. This was generated by determining an adjacency matrix for the set of base units using Geopandas. This adjacency matrix was then fed into a C++ implementation of Dijkstra's algorithm to compute the minimum number of edges that must be traversed in order to reach one unit from another. For more details on this methodology, see Appendix B.

3.2 Mathematical Model Results

Before considering the results of running this model on the state of Ohio, we first reference the current district plan in Ohio, as shown in Map 5. Notice that this plan includes districts that are widespread and have visibly awkward shapes. While the current districts manage to get within a margin of 1 person from the target population, it is clearly not compact, nor does it represent the political leanings of the state proportionally. Based on election results from the 2016 U.S. presidential election, 43.2% of Ohio's

voters vote Democrat and 51.3% vote Republican. However, 75% of the state's U.S. congressional districts have more Republican voters than Democrat voters. Thus, the map was clearly drawn with partisan goals in mind and aggressively violates proportionality.

Map 5

Map of Ohio's 2012-2022 U.S. Congressional District Plan

Note: Map reprinted from F. LaRose, "U.S. Congressional Districts 2012-2022 in Ohio," February 2018. [Online]. Available: https://www.ohiosos.gov/globalassets/publications/maps/2012-2022/congressional_2012-2020_districtmap.pdf. [Accessed 17 April 2021].

3.2.1 Dataset 1 Results

First, the model was tested on Dataset 1 as shown in Map 3, a map of county-level units where the 3 densest counties were split manually into sub-units of roughly equal area. In all experiments we used the solver Gurobi, either as hosted on the NEOS server or within a local installation of AMPL [17], [18], [19], [20], [21]. In all experiments we used an ε value of 1. We tested the model using a variety of settings for the maximum distance parameter, $maxDist$, as well as the limit on cut edges, $maxCut$. We treated these as model hyperparameters and effectively ran a grid search on values of $maxDist \in [5,6,7,8]$ and values of $maxCut$ between 100 and 130 in increments of 5 to find the most compact results possible. Since each of these results ended up yielding the same objective value, we choose the two most compact results to summarize here.

Each of these best experiments use a $maxCut$ value of 110, the minimum value given to the model that still returned a result. One experiment used a $maxDist$ value of 5 while the other used 6. Each of their results are summarized in Table 1. In the table we include the number of enclaves as defined by the number of units that border only one other unit of the same district (for districts with 3 or more units). When computing enclaves and cut edges for the state's current plan, we define any county or subsection of

a county (as divided by the state's district lines) to be its own unit. It is important to note these computed values for the state's current plan are not *perfectly* comparable since our plans are built from different basic-level units—but still provide a decent depiction.

Table 1

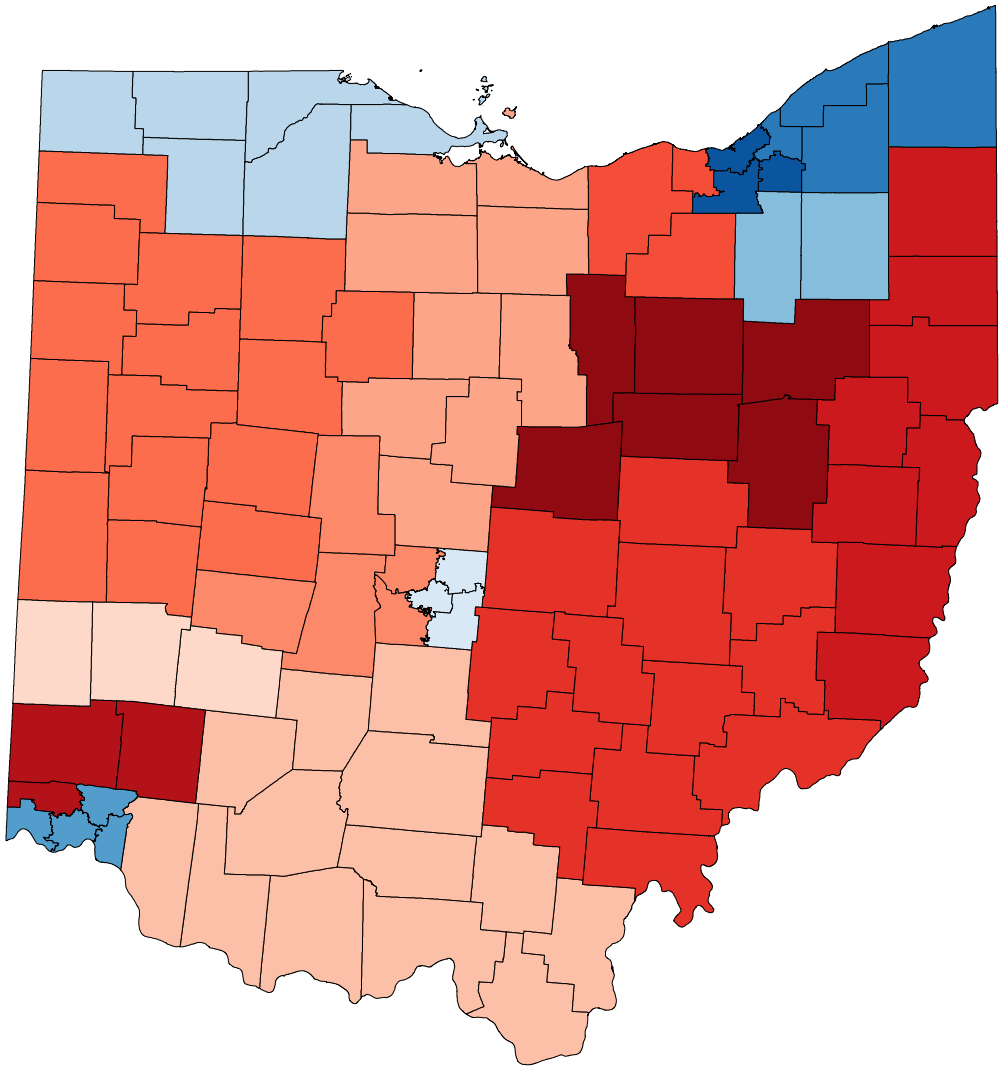
Summary of Mathematical Model's Best Results on Dataset 1

Map	$maxDist$	Cut Edges	Enclaves	Max Population Deviation	Dem/Rep Districts (Target Dem/Rep)
Map 6	5	110	10	7.3%	6/10 (7/9)
Map 7	6	109	10	7.3%	6/10 (7/9)
Current Plan, Map 5	10	142	19	<0.1%	4/12 (7/9)

The corresponding maps to these results follow in Map 6 and Map 7. Note that in each of the maps, districts colored with a reddish hue lean Republican while districts with a bluish hue lean Democrat.

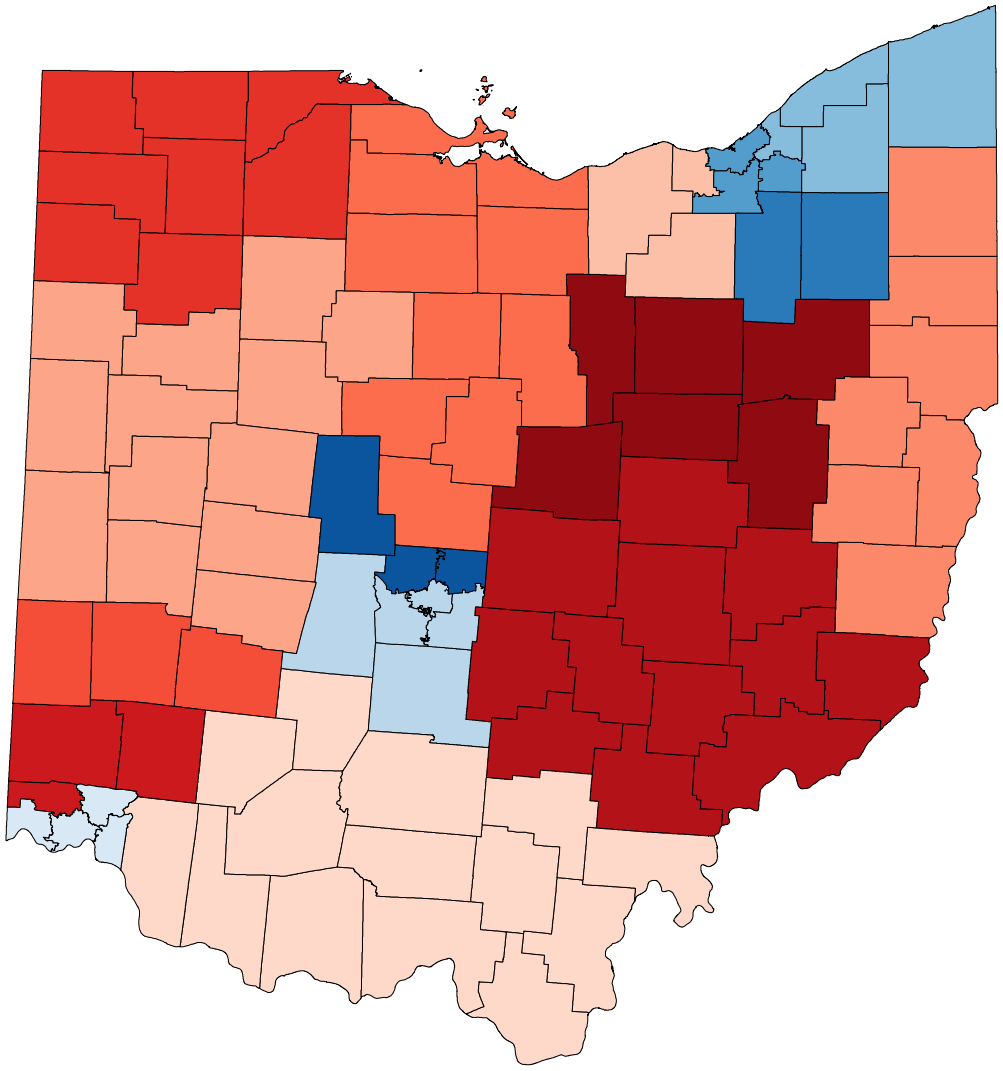
Map 6

Map of Model's Result on Dataset 1 with $maxDist = 5$, $maxCut = 110$, *and* $epsilon = 1$



Map 7

Map of Model's Result on Dataset 1 with $maxDist = 6$, $maxCut = 110$, *and* $epsilon = 1$



As shown, the model is clearly outputting a contiguous, sensible district map for the state of Ohio. On the account of proportionality, it meets the $targdem \pm epsilon$ threshold as required by the model, far beating the current plan on proportionality. In terms of compactness, both maps pass the eyeball test very well as districts appear quite compact, far better than the current plan. Map 7, which allows the $maxDist$ to be 1 greater, actually output a map with 1 fewer cut edge. For further comparison on the impact of the compactness constraints, see the additional results in Appendix C.

It is only on the account of maximum population deviation that our model struggles to compete with the current plan. The current districts manage to get within 1 person of their target population while ours has a maximum deviation of 7.3%. Since minimizing population deviation was our objective and these were optimal results—this is clearly a side effect of the base data layer of units fed into the model. Upon closer inspection of all our experimental results, it was found that they all actually reached the same objective value. This was due to the way the units were chosen in Cuyahoga county. All maps yielded the same 3-unit district entirely contained within Cuyahoga county and this district, while optimal for the plan, was the limiting factor for minimizing population deviation.

3.2.2 Dataset 2 Results

Since all results using Dataset 1 yielded the same objective value, Dataset 2 was created with the goal of using a similar number of units, while being extra careful to make them as even in population as possible. This was achieved by dividing 6 more of

the dense counties into sub-units and more carefully choosing the boundaries of these sub-units based on the amount of population contained within.

We tested the model on this dataset under similar conditions as we tested Dataset 1, but found the computation to take significantly longer and were unable to retrieve an optimal result from the solver Gurobi. This suggests that with Dataset 1 the model was able to find an optimal solution sooner due to the spurious patterns related to population totals within units—especially around Cuyahoga county.

Thus, in order to still see what types of plans could be output by the model under this dataset, we gave Gurobi the option to return a result early once it had reached a certain value in the objective. We kept $epsilon = 1$ and tested various options for this early stopping value as well as values of $maxDist$ and $maxCut$. Our best result occurred with $maxDist = 6$, $maxCut = 110$, and an early stopping value of $z = 19000$. We provide our best result here in Table 2 and Map 8 for comparison with the outputs using Dataset 1 and the state's current plan.

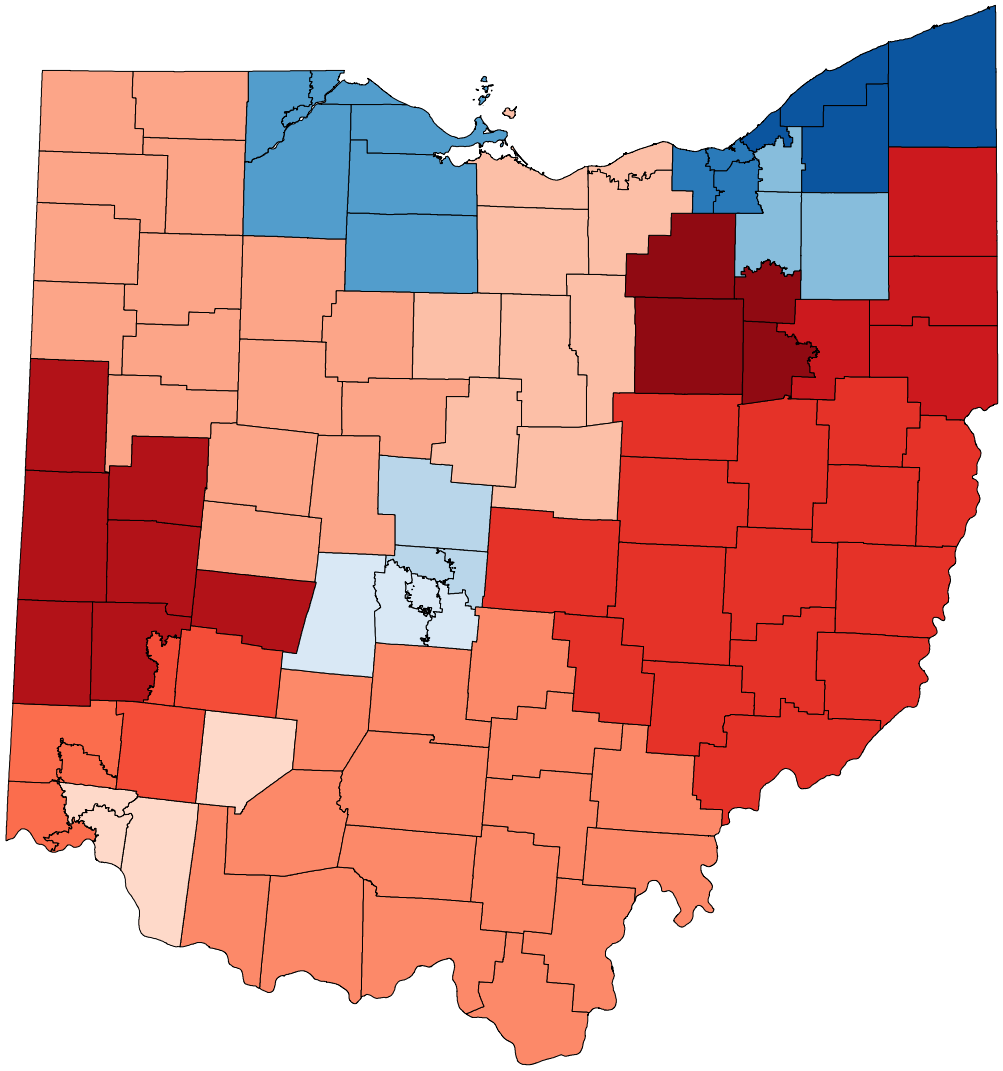
Table 2

Summary of Mathematical Model's Best Results on Dataset 2

Map	$maxDist$	Cut Edges	Enclaves	Maximum Population Deviation	Dem/Rep Districts (Target Dem/Rep)
Map 8	6	109	10	5.6%	6/10 (7/9)
Current Plan, Map 5	10	142	19	<0.1%	4/12 (7/9)

Map 8

Map of Model's Result on Dataset 2 *with* $maxDist = 6$, $maxCut = 110$, $epsilon = 1$, *and an early stopping value of* $z = 19000$



Clearly this result meets the goals of proportionality and compactness just as well as both of the results from using Dataset 1 while also reducing the maximum population deviation from 7.3% to 5.6%. This demonstrates the capability for the model to produce more optimal districts with respect to population deviation while highlighting that the limiting factor here is in the granularity of units we are able to provide the model. Providing more, smaller units could give the model greater flexibility to reduce the objective value, but this appears to become computationally expensive quickly.

Chapter 4: Future Directions and Summary

Based on these computational results, the model is doing a very good job of producing compact and contiguous districts that achieve proportionality, especially in comparison to Ohio's current districts. However, there is clearly room for improvement in several of the redistricting criteria discussed in Section 1.1 as well as in the model's overall complexity. Here we provide some suggestions for several areas of future work to extend this project.

4.1 Model Complexity

There is work to be done to further analyze the complexity of the model in terms of the number of decision variables, parameters, and constraints based on the units provided to the model. In particular, this analysis is necessary to begin considering ways to reduce the model's complexity. The reason for this is to better enable the model to return effective results on datasets with more units than those tested in this paper. Running the model on datasets with units at a higher granularity is clearly the key to

achieving smaller population deviations and if additional computational resources aren't an option, then a model with fewer variables or constraints could enable this.

In particular, our constraints related to both contiguity and compactness are costly. The cut edge constraint is implemented in a way that requires w_{ikj} auxiliary variables for every pair of neighboring units and every district in the state. Simply to achieve contiguity, we could require as many as $N^2 * M$ constraints. Could either of these criteria be met using a smaller model?

4.2 Partisan Fairness

Currently, the model uses proportionality to ensure that a resulting district plan is fair in a partisan sense. However, there are many other popular metrics used to measure partisan fairness. In the future, one could try to model and evaluate metrics like partisan symmetry or the efficiency gap with this model, as these metrics are explained in Section 1.3.

4.3 Compactness

In section 1.2, we give a wide variety of popular metrics for compactness. In our model we chose cut edges as our means of constraining for compactness due to its flexibility and simplicity in a number of geographic situations. However, it could be interesting to incorporate one or more of the other measures of compactness for comparison. Perhaps a different measure, or multiple measures together could yield a more compact plan. Perhaps using a different measure could result in similarly compact results but with fewer decision variables. Similarly, to promote compactness, constraints could be added to limit the number of enclaves, or units that only border one other unit in

the same district. See Appendix D for a modeling technique we propose using to constrain this but did not have time to evaluate in this project.

4.4 Integrity

Another key criterion of good districts was integrity where we aim to preserve other geographic subdivisions like counties, cities, townships, and communities of interest. Our current results achieve this fairly well by using basic-level units that are primarily chosen to be counties in the first place. If a future person were able to test the model on datasets with finer granularity like precincts, modeling would be necessary to limit the number of subdivisions, such as counties, that get split between districts.

4.5 Competitiveness

While this model goes out of the way to promote partisan proportionality, competitiveness is one criterion that we did not end up including in the model but could be considered for a future one. A future work could consider how best to define and constrain for competitiveness. Should all districts be made as competitive as possible and have partisan leanings be within a certain margin of one another? Should a certain number of districts be made as competitive as possible while the rest are proportionally made ‘safe’ districts for each particular party? Since the geographic concentration of certain voters tends to vary widely throughout a state (for example Democrats being concentrated in larger cities), this could be a difficult criterion to achieve while maintaining compactness. Further thought to the existing policy and fairness surrounding competitiveness should be taken and then this criterion could be evaluated as an additional modeling component.

4.6 Voting Rights Act

While partisan gerrymandering is still legal to a certain extent in many states, racial gerrymandering is explicitly prohibited by the Voting Rights Act. This current model does not take into account the demographics of a state nor the ability for minority groups to elect a candidate of choice. This type of analysis is essential when designing a new district plan in order to comply with the VRA and have a fair map. Considering how to use a model like this one to combat racial gerrymandering is a major future direction that we see as a priority for the model going forward.

4.7 Additional Experimentation

Beyond continuing to improve the model's performance under a variety of criteria, a major area for future work is in simply applying and evaluating the model in a variety of other situations. We offer the state of Ohio as a proof of concept, but the model should be applied and evaluated in other states with their own unique geographic and political characteristics. How do the results compare when applied to larger states like Texas? What type of results would the model return in a state like Iowa which requires basic-level units to be counties in the first place?

Furthermore, the model should not only be applied to other states but could also be applied to other levels of government. Here we consider the model being used to generate congressional districts, but perhaps it is better suited to produce a state's legislative districts which tend to have more lenient requirements on population balance. How does the model perform on situations like this? What might need tweaked?

4.8 Alternative Modeling Approaches

In the process of applying the model to other states and types of districts, the model is likely to need adjusted. As a future direction, one could consider other ways to more significantly restructure the model. Should population balance actually be the objective? Perhaps this objective could be made a constraint and the model could minimize for compactness. It would also be interesting to consider a multi-objective approach to the model. Could compactness and population balance be combined in the objective function? How should each goal be weighted?

4.9 Summary and Impact

This project ultimately acts as a proof of concept for applying integer linear programming-based techniques to optimize the process of redistricting. We are able to create a mathematical model that prioritizes population balance while also forcing that a state's district plan is *compact* and *fair*. We achieve impressive results on the state of Ohio, especially when compared to the state's current, heavily gerrymandered plan. This exhibits a strong potential for an ILP-based optimization model to have a seat at the redistricting table and be used to help generate fairer districts. There are a variety of other technologies and techniques used to help combat gerrymandering, but this type of model has significant potential to serve as yet another tool in a map drawer's tool belt.

However, it is important to note that it is not necessarily the intention of this paper to propose that algorithmic redistricting using a model like this one replaces the current redistricting processes where mapmaking commission are convened, public input is taken, and inappropriate districting decisions are challenged in court. Creating districts is

an inherently complex problem where computers stand to offer significant assistance—but it is also an inherently *human* problem. It is virtually impossible to incorporate and properly prioritize all of the human and geographic nuances that are particular to a certain state in a one-size-fits-all application. Human commissions and public input are ultimately necessary to properly represent a state's electorate. While we want to eliminate partisan gerrymandering from the redistricting process, redistricting is an inherently political decision and must be kept as such. The solution space for a state's district plan is *extremely* large with many viable solutions under a given set of criteria, so humans are required to be in the loop to help make that final decision on a truly optimal solution among many theoretically optimal solutions.

That said, with models like this one, mapmakers have a much greater ability to explore the solution space of redistricting and are able to find optimal possibilities under any given set of chosen criteria. We see this model being a significant tool to explore what solutions are possible when various criteria are optimized and constrained for simultaneously. With that goal in mind, this model has the ability to make a significant impact on the redistricting process in the future and we are optimistic that the country is moving towards a brighter future of *fair* redistricting.

Bibliography

[1] A. Blake, "Redistricting, Explained," The Washington Post, 1 June 2011. [Online]. Available: www.washingtonpost.com/politics/redistricting-explained/2011/05/27/AGWsFNGH_story.html?utm_term=.9a20b654409b. [Accessed 15 April 2021].

[2] "Redistricting Criteria," National Conference of State Legislators, 23 April 2019. [Online]. Available: https://www.ncsl.org/research/redistricting/redistricting-criteria.aspx. [Accessed 15 April 2021].

[3] *VRA. 52 U.S.C. § 10301 et seq.,* 2020.

[4] D. D. Polsby and D. R. Popper, "The Third Criterion: Compactness as a Procedural Safeguard Against Partisan Gerrymandering," *Yale Law and Policy Review,* vol. 9, no. 2, 1991.

[5] E. C. Reock, "A Note: Measuring Compactness as a Requirement of Legislative Apportionment," *Midwest Journal of Political Science,* vol. 5, 1961.

[6] A. Adhia, D. Grozdanov, M. Ramezanzadeh and Z. Fisher, "Measuring Compactness," [Online]. Available: https://fisherzachary.github.io/public/r-output.html. [Accessed 15 April 2021].

[7] M. Duchin and B. E. Tenner, "Discrete geometry for electoral geography," *arXiv:1808.05860 [physics]*, 15 August 2018. [Online]. Available: https://arxiv.org/abs/1808.05860. [Accessed 2021 April 17].

[8] D. DeFord and M. Duchin, "Redistricting Reform in Virginia: Districting Criteria in Context," *Virginia Policy Review,* vol. 12, no. 2, pp. 120-146, 2019.

[9] N. Stephanopoulos and E. McGhee, "Partisan Gerrymandering and the Efficiency Gap," *The University of Chicago Law Review,* vol. 82, no. 2, pp. 831-900, 2015.

[10] M. Bernstein and M. Duchin, "A Formula Goes to Court: Partisan Gerrymandering and the Efficiency Gap," *Notices of the American Mathematical Society,* vol. 64, no. 9, pp. 1020-1024, 2017.

[11] G. King, B. Grofman, A. Gelman and J. Katz, "Brief of Amici Curiae Professors Gary King, Bernard Grofman, Andrew Gelman, and Jonathan Katz in Support of Neither Party," *U.S. Supreme Court in Jackson v. Perry,* 2005.

[12] D. DeFord, N. Dhamankar, M. Duchin, V. Gupta, M. McPike, G. Schoenbach and K. W. Sim, "Implementing partisan symmetry: Problems and paradoxes," *arXiv:2008.06930 [physics]*, 3 March 2021. [Online]. Available: http://arxiv.org/abs/2008.06930. [Accessed 2021 April 17].

[13] A. Becker and J. Solomon, "Redistricting Algorithms," *arXiv:2011.09504 [cs]*, 18 November 2020. [Online]. Available: http://arxiv.org/abs/2011.09504. [Accessed 2021 April 17].

[14] D. DeFord, M. Duchin and J. Solomon, "Recombination: A Family of Markov Chains for Redistricting," *arXiv:1911.05725 [physics]*, 31 October 2019. [Online]. Available: http://arxiv.org/abs/1911.05725. [Accessed 17 April 2021].

[15] R. Buck, K. Jolly and K. Kelly, "Statewide Shapefile for Ohio: Precincts 2016," Metric Gerrymandering and Geometry Group, 14 November 2019. [Online]. Available: https://github.com/mggg/ohio-precincts. [Accessed 14 September 2020].

[16] S. Manson, J. Schroeder, D. Van Riper, T. Kugler and S. Ruggles, "IPUMS National Historical Geographic Information System: Version 15.0 [dataset]," IPUMS, Minneapolis, MN, 2020, doi: 10.18128/D050.V15.0.

[17] Dolan and E. D, "NEOS Server 4.0 Administrative Guide," *arXiv:cs/0107034 [cs]*, July 2001. [Online]. Available: http://arxiv.org/abs/cs/0107034. [Accessed 14 Sepetember 2020].

[18] W. Gropp and J. J. More, "Optimization environments and the NEOS server," Argonne National Lab., IL (United States), ANL/MCS-P-654-0397; CONF-9607197-1, March 1997. [Online]. Available: https://www.osti.gov/biblio/563264-optimization-environments-neos-server. [Accessed 14 Sepetember 2020].

[19] J. Czyzyk, M. P. Mesnier and J. J. More, "The NEOS Server," *IEEE Computational Science and Engineering*, vol. 5, no. 3, pp. 68-75, July 1998, doi: 10.1109/99.714603.

[20] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," *arXiv:cs/0107034 [cs]*, 2021. [Online]. Available: http://www.gurobi.com. [Accessed 2020 September].

[21] R. Fourer, D. M. Gay and B. W. Kernighan, *AMPL: a modeling language for mathematical programming*, 2nd ed. Pacific Grove, CA: Thomson/Brooks/Cole, 2003.

[22] F. LaRose, "U.S. Congressional Districts 2012-2022 in Ohio," February 2018. [Online]. Available: https://www.ohiosos.gov/globalassets/publications/maps/2012-2022/congressional_2012-2020_districtmap.pdf. [Accessed 17 April 2021].

Appendix A: AMPL Implementation of Mathematical Model

```
param N; #number of units in the state

param M; #number of electoral districts

set Units; #the set of state units

set Districts := 1..M;  #the set of districts

param p{i in Units, j in Districts} binary;
#is 1 if unit i can be included in district j

set possiblePairs := {i in Units, j in Districts: p[i,j] == 1};
#set of possible unit/district pairings

param dist{i in Units, k in Units} integer;
#distance between units i and j measured by how many adjacent units away i is
from j

param maxDist; #maximum allowable distance between two units in a district

param maxCut; #maximum number of cut edges in district plan

param dem{i in Units}; #the number of Democrat voters in a unit

param rep{i in Units}; #the number of Republican voters in a unit

param vot{i in Units} := dem[i] + rep[i];
#the number of total Democrat and Republican voters in unit

param LARGE := sum{i in Units}vot[i]; #sum of all voters in state

param targDem := round(((sum{i in Units}dem[i]) / (sum{i in Units}vot[i]))*M);
#proportion of Democrat voters multiplied by total number of districts

param targPop := round((sum{i in Units}vot[i]) / M);

param epsilon; #a tuning parameter for the allowable error of actual Democrat
districts from the ideal

var y{j in Districts} binary;
#is 1 if more Democrats in a district, 0 if not
```

```
var s{i in Units, k in Units: dist[i,k] == 1} binary;
#is 1 if neighboring units i and j are in the same district

set possibleSameDistrict{i in Units, k in Units: dist[i,k] == 1} :=
    {j in Districts: p[i,j] == 1 and p[k,j] == 1};

var w{i in Units, k in Units, j in possibleSameDistrict[i,k]: dist[i,k] == 1}
    binary;
#auxiliary variable for cut edge constraint

var x{i in Units, j in Districts: p[i,j] == 1} binary;
#is 1 if unit i is chosen for district j

var z; #the largest difference from target population among all districts

minimize LargestDifference: z;

subject to zIsLargestDifferencePos{j in Districts}:
    z >= (sum{(i,j) in possiblePairs}x[i,j]*vot[i]) - targPop;

subject to zIsLargestDifferenceNeg{j in Districts}:
    z >= targPop - (sum{(i,j) in possiblePairs}x[i,j]*vot[i]);

subject to Exactly1DistrictPerUnit{i in Units}:
    sum{(i,j) in possiblePairs}x[i,j]=1;

set bridgeUnits{k in Units, l in Units, d in 2..maxDist: dist[k,l] == d} :=
    {i in Units: (dist[i,k] == 1 and dist[i,l] == d - 1) or
                 (dist[i,k] == d - 1 and dist[i,l] == 1)};

subject to unitBridge{k in Units, l in Units, j in Districts, d in 2..maxDist:
                      dist[k,l] == d and p[k,j] == 1 and p[l,j] == 1}:
    sum{i in bridgeUnits[k,l,d]: p[i,j] == 1}x[i,j] >= x[k,j] + x[l,j] - 1;

subject to setOneIfLeansDemocrat{j in Districts}:
    LARGE * y[j] >= sum{(i,j) in possiblePairs}(x[i,j]*dem[i] - x[i,j]*rep[i]);

subject to setZeroIfLeansRepublican{j in Districts}:
    LARGE * (1 - y[j]) >= sum{(i,j) in possiblePairs}(x[i,j]*rep[i] -
                                                      x[i,j]*dem[i]);

subject to lessThanMaxDemocratDistricts:
    sum{j in Districts}y[j] <= targDem + epsilon;

subject to greaterThanMinDemocratDistricts:
```

```
    sum{j in Districts}y[j] >= targDem - epsilon;

subject to maxDistance{i in Units, k in Units, j in Districts:
                       p[i,j] == 1 and p[k,j] == 1 and dist[i,k] > maxDist}:
    x[i,j] + x[k,j] <= 1;

subject to setSSameDistrict{i in Units, k in Units, j in Districts:
                            dist[i,k] == 1 and p[i,j] == 1 and p[k,j] == 1}:
    s[i,k] >= x[i,j] + x[k,j] - 1;

subject to setSDifferentDistrict{i in Units, k in Units, j in Districts:
                                 dist[i,k] == 1 and p[i,j] == 1 and
                                 p[k,j] == 1}:
    s[i,k] <= 0.5 * (x[i,j] + x[k,j]) + (1 - w[i,k,j]);

subject to setY{i in Units, k in Units: dist[i,k] == 1}:
    sum{j in possibleSameDistrict[i,k]}w[i,k,j] >= 1;

subject to maxCutEdges:
    sum{i in Units, k in Units: dist[i,k] == 1}(1-s[i,k]) <= maxCut;
```

Appendix B: Using Dijkstra's Algorithm to Compute Unit Distances

A key component to the constraints of the model considered in this project is a 'distances' parameter. By defining a distance between any two units in a state, we can constrain our model to select districts that never have two units more than some maximum distance value apart. This distance parameter could be defined as the direct distance in miles between two units for example. In the case of our testing on the state of Ohio, we employ a distance defined in the following way due to its benefit for logical constraints regarding contiguity:

$$Distance(i,k) \text{ s.t. } i,k \in Units$$

$$= \text{minimum number of edges that must be traversed to reach j from i}$$

Thus, in order to generate this data, we can first obtain adjacency information for all the units in a state (i.e. for every pair of units we record whether they are adjacent to one another). Then we can create an instance of a graph from the units in the state where each node is defined to be a unit and there exists an edge between two nodes if the corresponding units are adjacent.

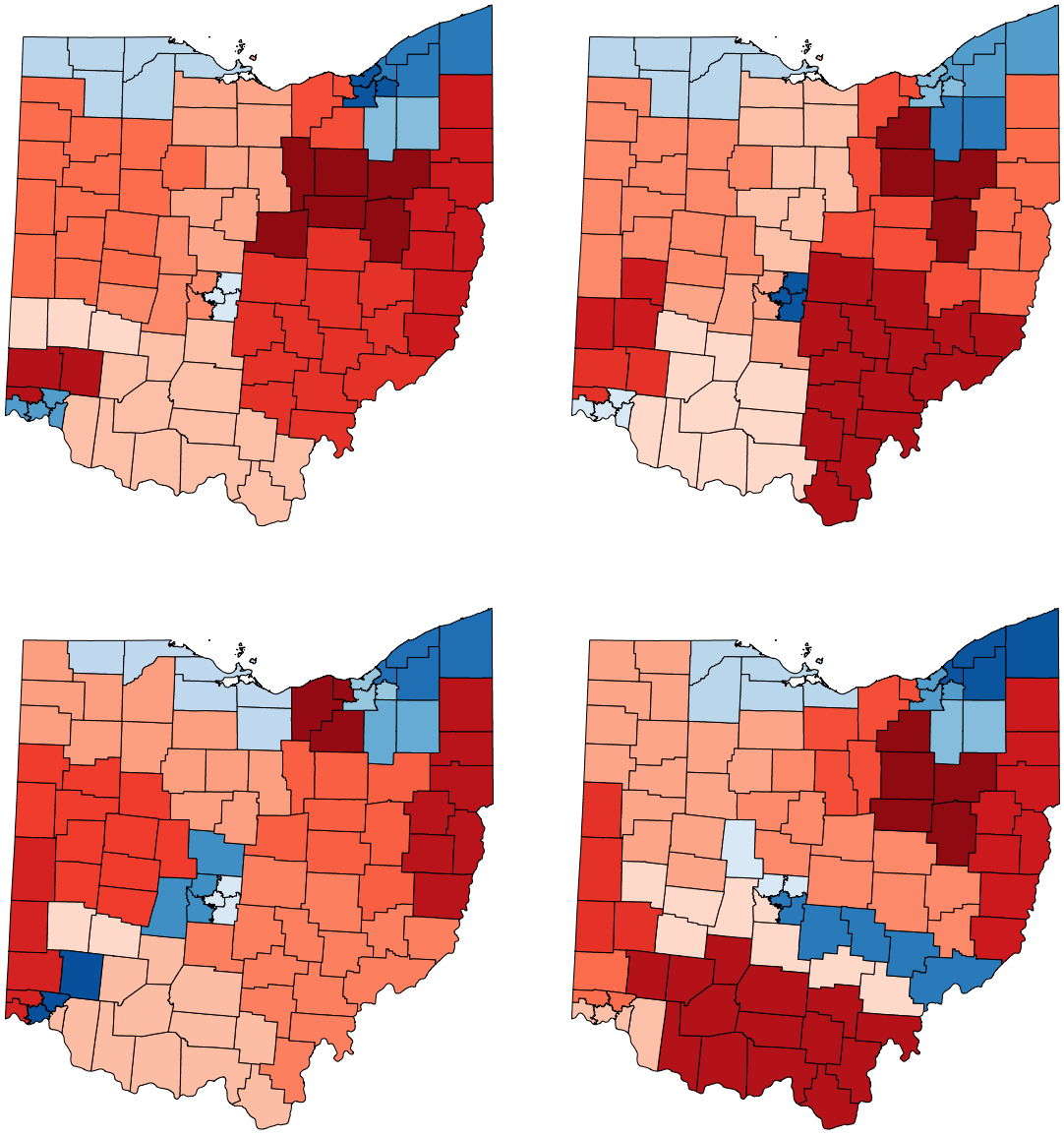
Using this graph structure modeling the state as a whole, we simply apply Dijkstra's shortest path algorithm to find the shortest path between every pair of nodes in the graph. The length of this shortest path between two nodes is then defined to be the distance between the nodes' corresponding units.

For more details on the implementation of this algorithm, see the project repository and its corresponding subdirectory on 'shortestDistances': https://github.com/benjamincarman/repairing-redistricting/tree/master/shortestDistances

Appendix C: Additional Results

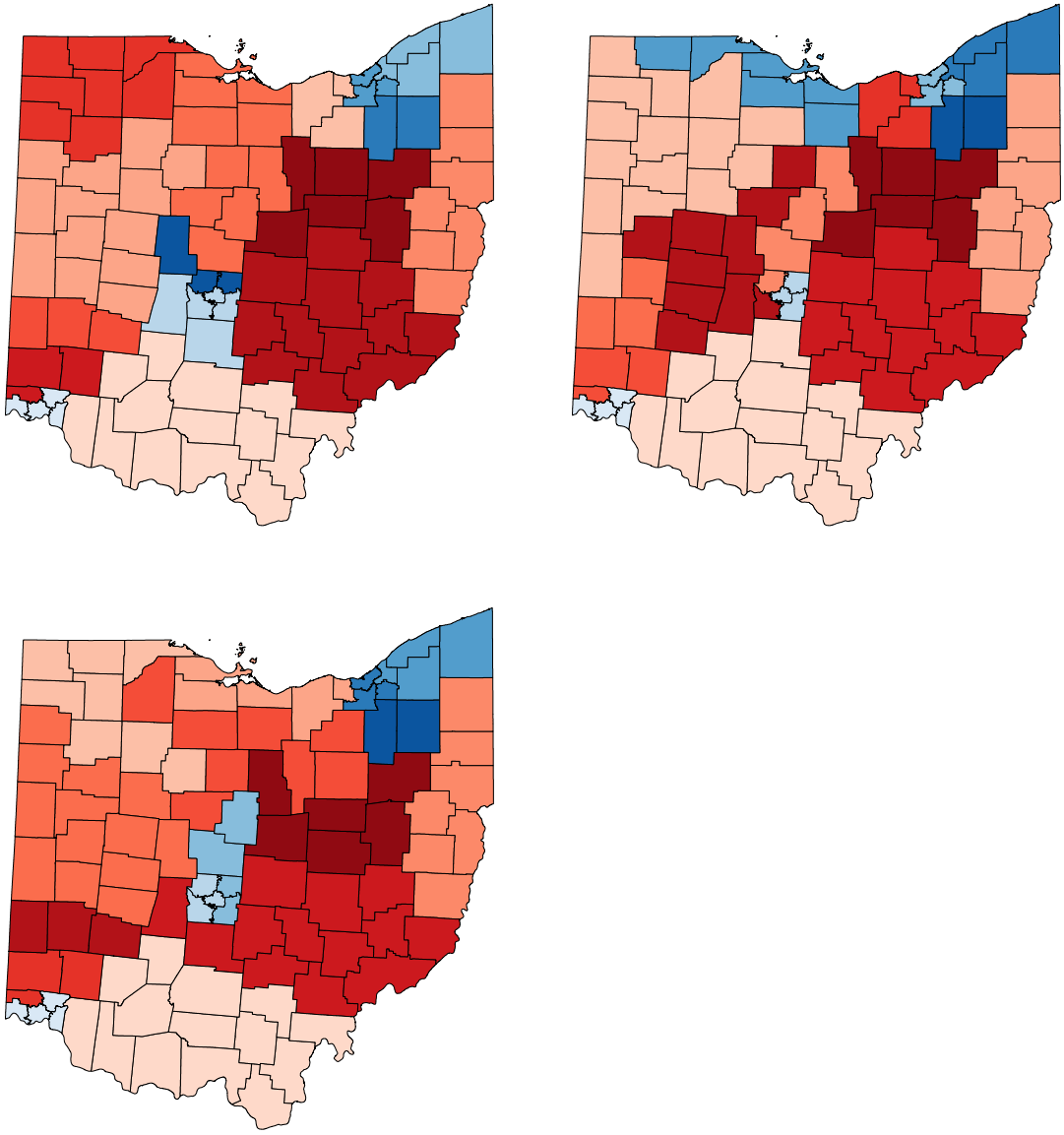
Here we provide a selection of additional computational results from Dataset 1. In particular, these results depict the impact had by our compactness constraints as inputs for $maxDist$ and $maxCut$ vary. All results reached the same objective value.

Results with $maxDist = 5$



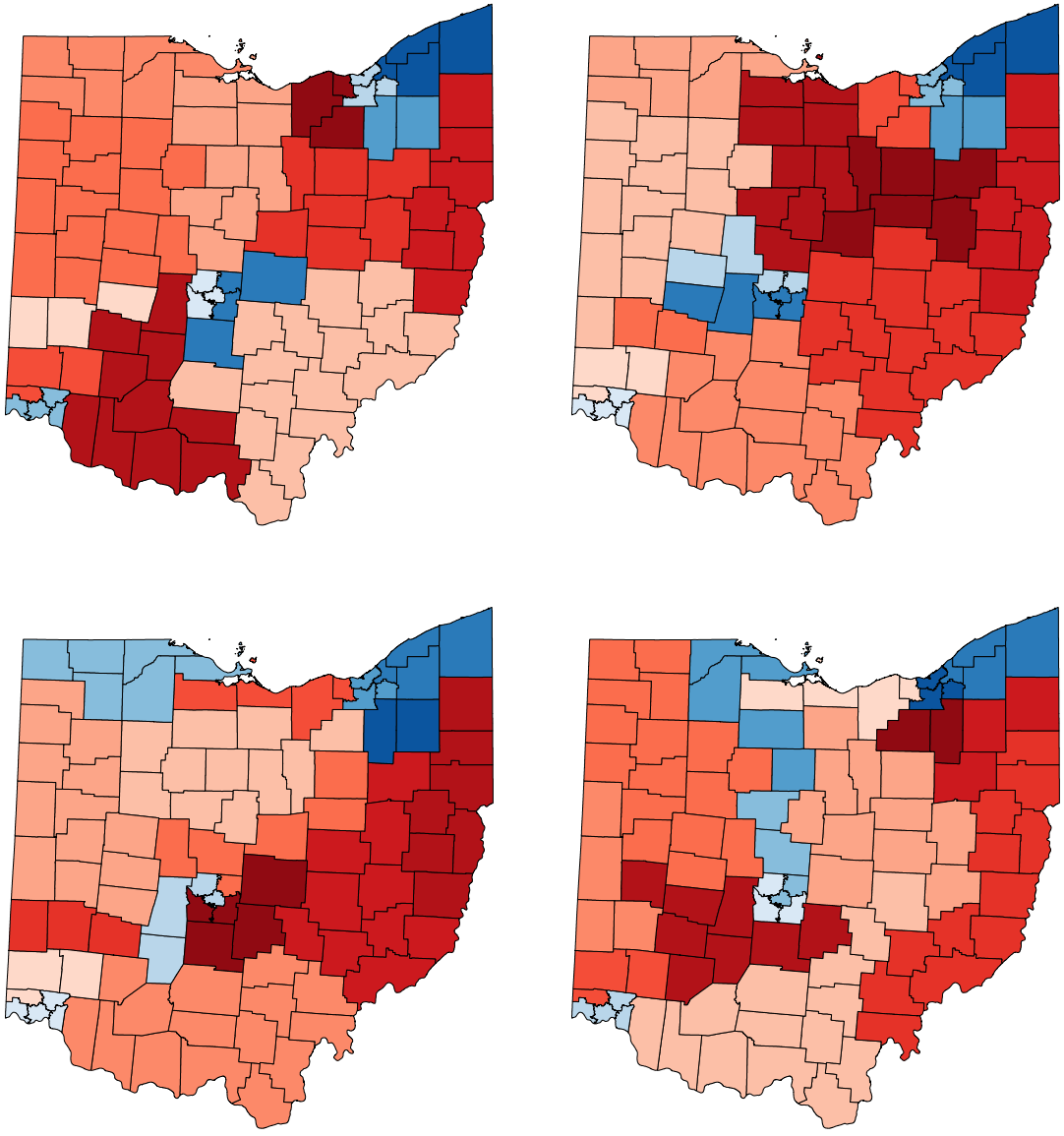
Cut Edges: Top-Left = 110, Top-Right = 115, Bottom-Left = 115, Bottom-Right = 125

Results with $maxDist = 6$



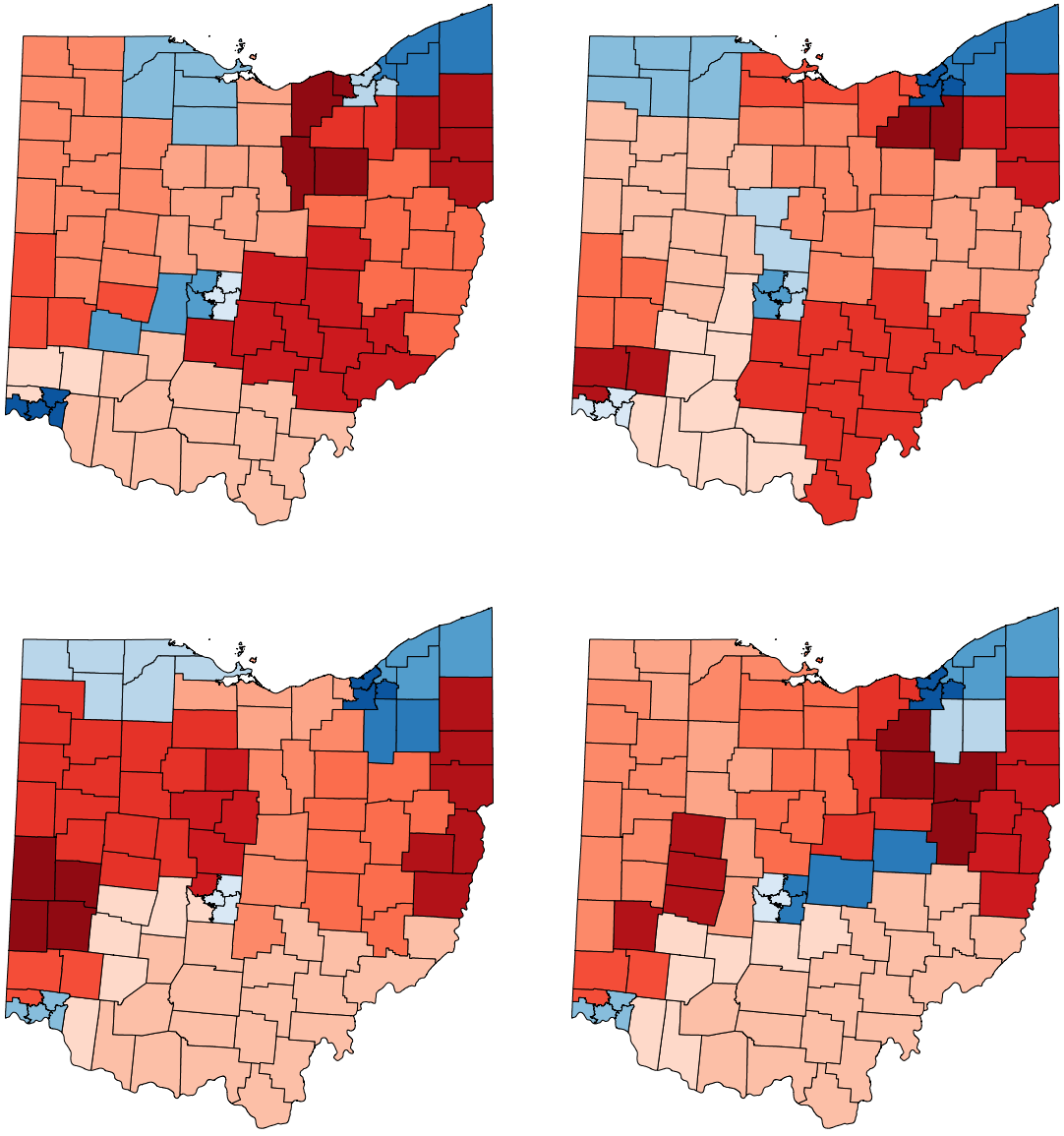
Cut Edges: Top-Left = 109, Top-Right = 113, Bottom-Left = 119

Results with $maxDist = 7$



Cut Edges: Top-Left = 110, Top-Right = 113, Bottom-Left = 119, Bottom-Right = 125

Results with $maxDist = 8$



Cut Edges: Top-Left = 110, Top-Right = 113, Bottom-Left = 118, Bottom-Right = 127

Appendix D: Modeling Technique for Preventing Enclaves

Here we propose a modeling technique to constrain enclaves—units that only border one other unit in the same district. We did not have time to evaluate and include this in the main model but provide it here as a resource for future extensions.

Let $maxEnclaves$ be the maximum number of enclaves allowed

Let $e_i = \begin{cases} 1, \text{if unit } i \text{ is an enclave} \\ 0, \text{otherwise} \end{cases}$

$$\forall\, i \in Units$$

$$\sum_{k \in Units:\, dist_{ik}=1} s_{ik} \geq 2 * (1 - e_i), \qquad \forall\, i \in Units \tag{C14}$$

$$\sum_{i \in Units} e_i \leq maxEnclaves \tag{C15}$$

This set of constraints works by generally forcing the sum of s_{ik} variables for a given Unit, i, to be at least 2, meaning Unit i must be in the same district as at least 2 of its neighbors. We multiply the right-hand side of (C14) by $(1 - e_i)$ thus placing no limit on s_{ik} if the model chooses Unit i to be an enclave. We can then limit this number of chosen enclaves by the value $maxEnclaves$ in (C15). Allowing a certain number of maximum enclaves this way could be necessary to allow for some 1-Unit and 2-Unit districts. A set of constraints like this one regarding enclaves could be a very beneficial future direction for the model.